Introduction to Neural Networks
U. Minn. Psy 5038

Gaussian generative models, learning, and inference

- **Initialize standard library files:**

```
Off[General::spell1];
```

Last time

Quick review of probability and statistics

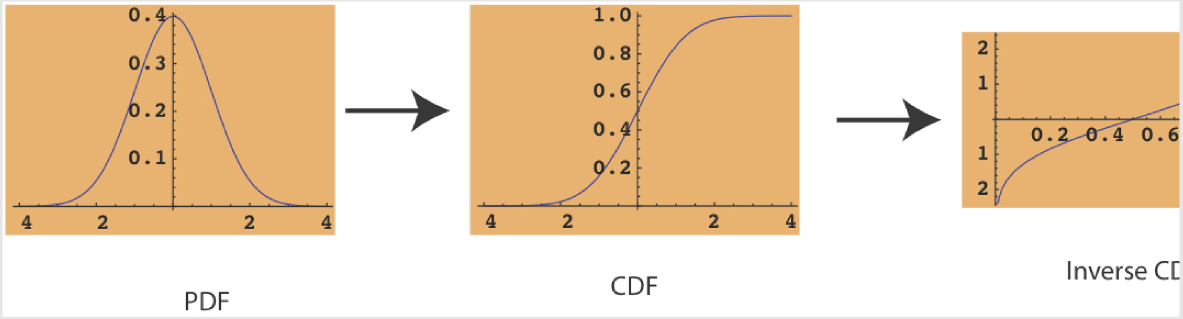
Applications to random sampling

If we know p(x), and are given a function, y=f(x), what is p(y)?

$$p_Y(y)\ \delta y = p_X(x)\ \delta x$$

This principle is used to make random number generators for general probability densities from the uniform distribution. The result is that one can make a random draw from a uniform distribution p(x), from between 0 and 1, and go

to the inverse CDF to read off the value of the random sample from p(y).



Today

Review of big picture

Examples of computations on continuous probabilities

Examples of computations on discrete probabilities

Introduction to Bayes learning

Recall relationship between "energy" neural networks and Bayesian inference

Here we are only talking about inference or estimation based on
patterns of neural activity –– i.e. in the language of neural networks, about " recall ",
rather than learning. Later we will introduce Bayesian learning.

In the general case we can talk about the probability over
all possible values of a neural network' s state vector : `p (V1, V2, ...)`

This doesn't distinguish which values are fixed and which are allowed to vary. If some values are fixed, then we can treat those as the input, and allow the network's free neurons to vary to maximize a conditional probability:

Relationship between posterior probability and the energy of the state of a Hopfield neural network:

$$p(V_1,\ldots,V_m|V_1^s,\ldots,V_k^s) = \kappa' \exp\left(\frac{-E\left(V_1,\ldots,V_m;V_1^s,\ldots,V_k^s\right)}{T}\right)$$

$$p(H|d)$$

H (the hypothesis space) corresponds to the values of state variables, i.e. patterns of neural activity that are changing $\{V_1, V_2,\ldots\}$. d corresponds to "data" or the fixed, clamped values $\{V_1^s,\ V_2^s\ldots\}$

From the more abstract point of view of statistical inference, we have some variables that are fixed--the "data" d, some we let vary to maximize probability--the hypotheses H, and some that we may not care to estimate (n). Let's see how to use these distinctions.

Suppose we have p(H,d,n). Two rules of inference:

Marginalize over what you don't care about:

$$p(H, d) = \sum_n p(H, d, n)$$

Condition on what you know:

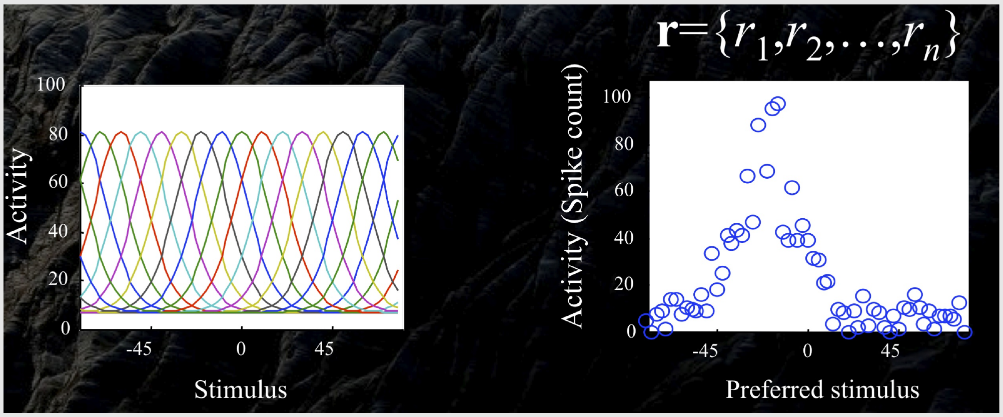
$$p(H \mid d) = \frac{p(H, d)}{p(d)}$$

Neural population codes and probability *distributions*

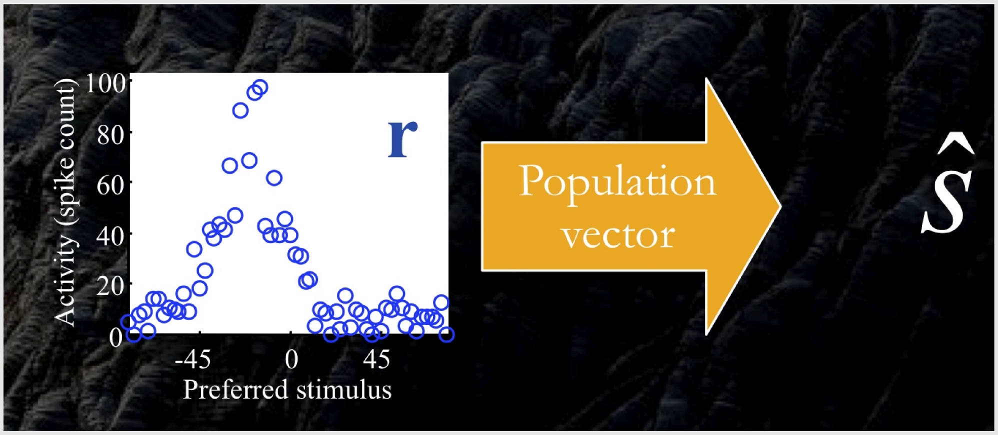
Energy methods show how a population of neurons could interact to compute a single value corresponding to the "most probable" solution.

Recent behavioral studies show that humans make decisions that combine information so as to take into account uncertainty. How might neural populations represent uncertainty? Probability distributions? (See Pouget et al., 2006 in the Readings).

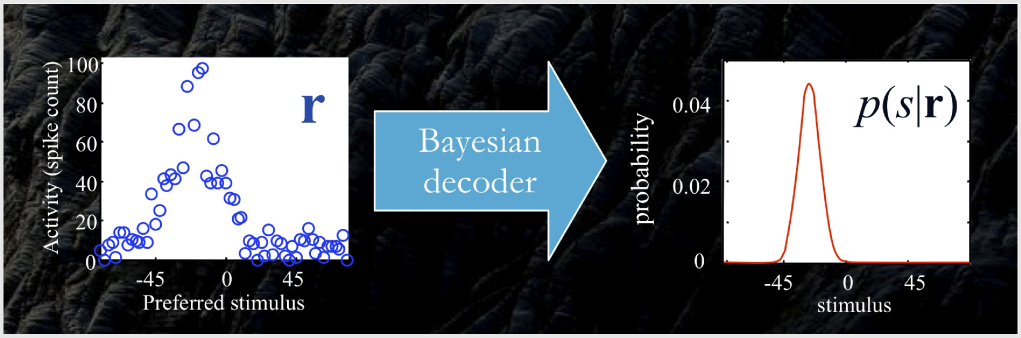
Recall the ideas of population codes from Lecture 16, where a stimulus attribute might be the orientation of a line, and the activity or spike count r_i, the response of the ith unit. Each unit i has a tuning function with a preferred orientation.



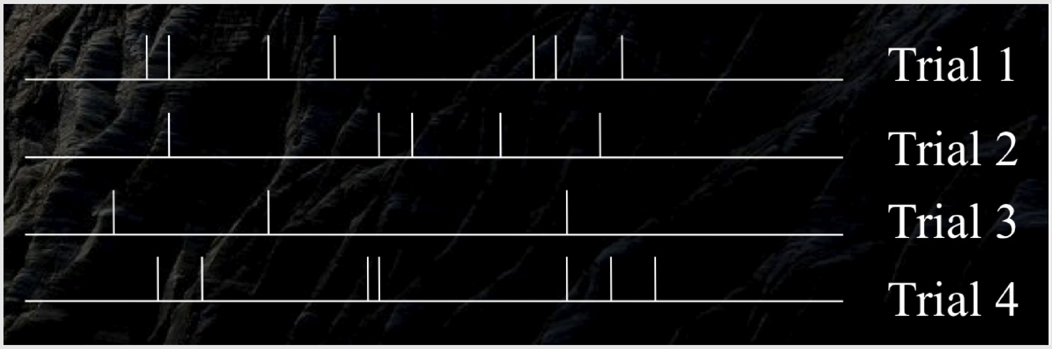
In lecture 16, we showed how the notion of a population vector has been applied to explaining a diverse range of phenomena including adaptation effects and motor planning, through the computation of a single estimate analogous to the center of mass. But do population codes represent just single values? E.g. suppose vector **r** repesents the pattern of spike counts over a population of orientation-tuned neurons. We saw how to estimate the value of orientation, $\hat{s}$, which that population represents.



But what if more information could be represented and used, that includes knowledge of the uncertainty--or more generally, the posterior distribution of s given **r**, p(s | **r**)?



Poisson model, p(**r** | s) is a reasonable first approximation to the variability that results in spike counts for repeated applications of the same stimulus. To compute with distributions requires a mechanism that can combine information from more than one distribution. Pouget and colleagues have shown that "poisson-like" distributions have a special status in that p(s | r1) p(s | r2) is proportional to p(s | r1 + r2).



Figures adapted from lecture by Alex Pouget.

Mathematica functions for gaussian multivariates & exploring marginals

Define PDF, CDF

MultinormalDistribution$[\mu, \Sigma]$ specifies *a* multinormal (multivariate Gaussian) distribution with mean vector μ and covar

In[253]:=

```
m1 = {1,1/2};
r=(1/2)*{{1,2/3},{2/3,4}};
ndist = MultinormalDistribution[m1, r];
pdf = PDF[ndist, {x1, x2}]
```

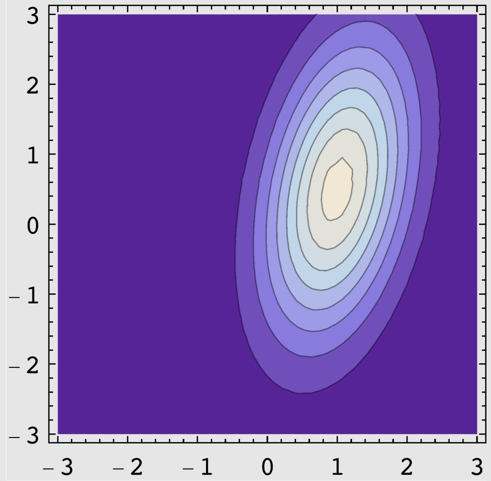
Out[256]=

$$\frac{3\, e^{\frac{1}{2}\left(-(-1+\text{x1})\left(\frac{9}{4}(-1+\text{x1})-\frac{3}{8}\left(-\frac{1}{2}+\text{x2}\right)\right)-\left(-\frac{3}{8}(-1+\text{x1})+\frac{9}{16}\left(-\frac{1}{2}+\text{x2}\right)\right)\left(-\frac{1}{2}+\text{x2}\right)\right)}}{4\sqrt{2}\,\pi}$$

In[257]:=

```
g1 = ContourPlot[PDF[ndist, {x1, x2}], {x1, -3, 3}, {x2, -3, 3}]
```

Out[257]=



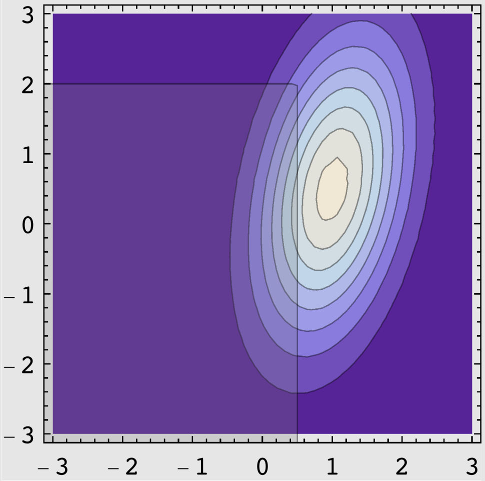
What is the probability of the distribution in the region $x_1 < .5 \bigcap x_2 < 2$?

In[240]:=

```
grp = RegionPlot[x1 < .5 && x2 < 2, {x1, -4, 4}, {x2, -4, 4},
   PlotStyle → Directive[Opacity[.25], EdgeForm[], FaceForm[Gray]]];

Show[{ g1, grp}, ImageSize → Small]
```

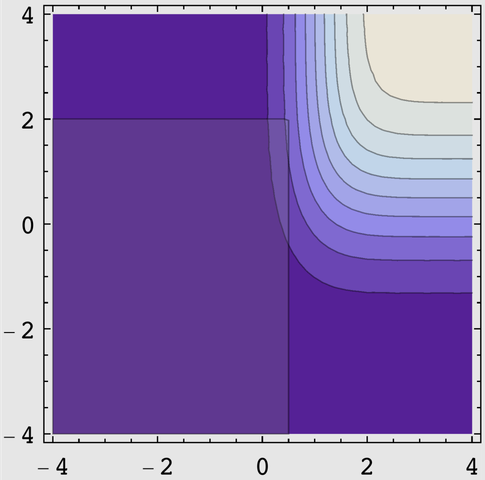
Out[241]=



In[242]:=

```
gcdf = ContourPlot[CDF[ndist, {x1, x2}], {x1, -4, 4}, {x2, -4, 4},ImageSize
Show[{ gcdf, grp}, ImageSize → Small]
```

Out[243]=



```
CDF[ndist, {.5, 2.0}]
```

```
0.225562
```

Calculating the marginals.

In[244]:=

```
Clear[x1, x2];
marginal[x1_] := ∫_{-∞}^{∞} PDF[ndist, {x1, x2}] ⅆx2;
marginal2[x2_] := ∫_{-∞}^{∞} PDF[ndist, {x1, x2}] ⅆx1;
```

In[247]:=

```
mt = Table[{x1, marginal[x1]}, {x1, -3, 3, .2}];

g2 = ListPlot[mt, Joined → True, PlotStyle → {Red, Thick}, Axes → False];
```

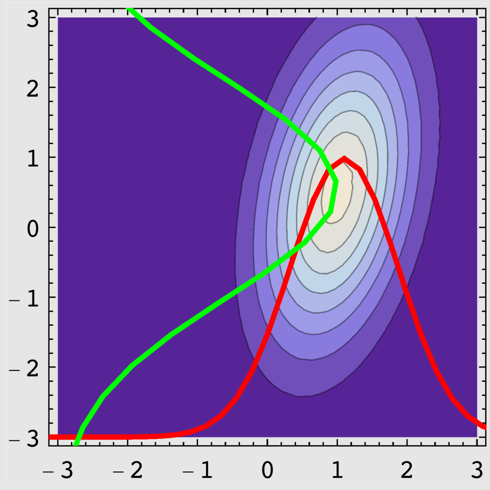
In[249]:=

```
mt2 = Table[{x2, marginal2[x2]}, {x2, -3, 3, .4}];
g3 = ListPlot[mt2, Joined → True, PlotStyle → {Green, Thick},
   Axes → False];
```

In[251]:=

```
theta = Pi / 2;
Show[g1,
 Epilog → {Inset[g2, {0, -3}, {0, 0}],
   Inset[g3, {-3, 0}, {0, 0}, Automatic,
    {{Cos[theta], Sin[theta]}, {Sin[theta], -Cos[theta]}}]}]
```

Out[252]=



■ Finding the mode

For the Gaussian case, the mode vector corresponds to the mean vector. But we can pretend we don't know that, and find the maximum and the coordinates where the max occurs:

```
In[260]:= FindMaximum[PDF[ndist, {x1, x2}], {{x1, 0}, {x2, 0}}]
```

```
Out[260]= {0.168809, {x1 → 1., x2 → 0.5}}
```

■ Drawing samples

As we've used in earlier lectures, drawing samples is done by:

```
RandomReal[ndist]
```

```
{1.07766, 0.280209}
```

■ Mixtures of gaussians with MultinormalDistribution[]

Multivariate gaussian distributions are often inadequate to model real-life problems, that for example might involve more than one mode. One solution is to approximate more general distributions by a sum or mixture of gaussians.

```
In[261]:= Clear[mix];
```

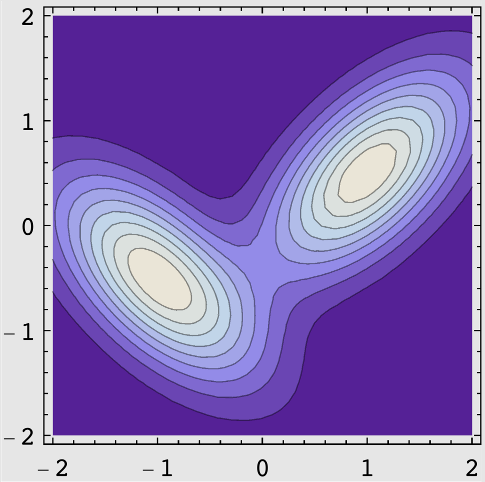
```
In[262]:= r1=0.4*{{1,.6},{.6,1}};
          r2=0.4*{{1,-.6},{-.6,1}};
          m1 = {1,.5}; m2 = {-1,-.5};
          ndist1 = MultinormalDistribution[m1, r1];
          ndist2 = MultinormalDistribution[m2, r2];
```

```
In[267]:= mix[x_] := 0.5 (PDF[ndist1, x] + PDF[ndist2, x]);
```

In[268]:=

```
gg1 = ContourPlot[mix[{x1, x2}], {x1, -2, 2}, {x2, -2, 2},
  PlotRange → Full, ImageSize → Small]
```

Out[268]=



■ Marginals for mixture

```
marginal[x1_] := Integrate[mix[{x1, x2}], {x2, -Infinity, Infinity}]        (1)
```

In[269]:=

```
Clear[marginal];
marginal[x1_] :=
  0.5 * (NIntegrate[PDF[ndist1, {x1, x2}], {x2, -Infinity, Infinity}] +
     NIntegrate[PDF[ndist2, {x1, x2}], {x2, -Infinity, Infinity}]);
```

In[271]:=

```
gg2 = Plot[marginal[x1], {x1, -2, 2}, PlotStyle → {Red, Thick},
   Axes → {False, False}];
```

In[272]:=

```
Clear[marginal];
marginal[x2_] :=
  0.5 * (NIntegrate[PDF[ndist1, {x1, x2}], {x1, -Infinity, Infinity}] +
     NIntegrate[PDF[ndist2, {x1, x2}], {x1, -Infinity, Infinity}]);
```

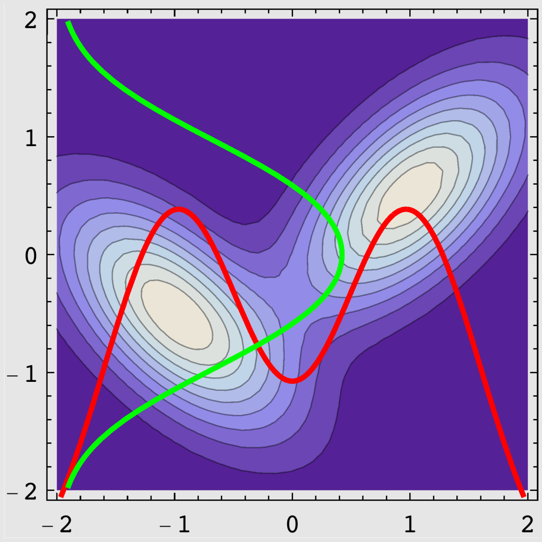
In[274]:=

```
gg3 = Plot[0.0333 * marginal[x2], {x2, -2, 2}, PlotStyle → {Green, Thick},
   Axes → {False, False}];
```

In[275]:=

```
theta = Pi / 2;
Show[gg1, ImageSize → 200,
 Epilog → {Inset[gg2, {0, -3}, {0, 0}],
   Inset[gg3, {-2, 0}, {0, 0}, Automatic,
    {{Cos[theta], Sin[theta]}, {Sin[theta], -Cos[theta]}}]}]
```

Out[276]=



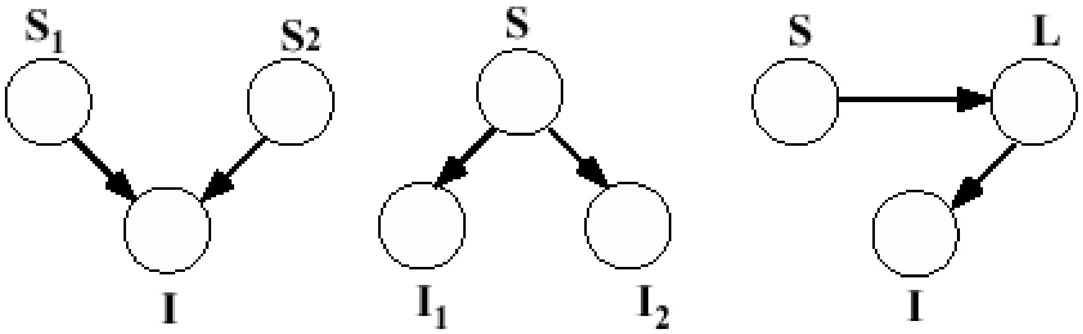
Which projection (marginal) is more "interesting"--the one onto x1 or onto x2?

Exploratory projection pursuit

Graphical Models of dependence

Graphs: causal structure and conditional independence

In general, natural patterns are specified by a high-dimensional joint probability, requiring a complex conditional relationships. The idea is to represent the probabilistic structure of the joint distribution P(S,L,I) by a Bayes net (e.g. Ripley, 1996}, which is a graphical model that expresses how variables influence each other. There are three basic building blocks: converging, diverging, and intermediate nodes. For example, multiple (e.g. scene) variables causing a given image measurement, a single variable producing multiple image measurements, or a cause indirectly influencing an image measurement through an intermediate variable. These types of influence provide a first step towards modeling the joint distribution and the means to compute probabilities of the unknown variables given known values.



Components of the generative structure for image patterns involve converging, diverging,and intermediate nodes. For example,these could correspond to:multiple (scene) causes {shape S1, illumination S2 giving rise to the same image measurement, I ; one cause, S influencing more than one image measurement, {color, I1, brightness, I2}; a scene (or other) cause S, {object identity, S} influencing an image measurement (image contour) through an intermediate variable L (3D shape) .

The arrows tell us how to factor the joint probability into conditionals. So for the three examples above, we have:

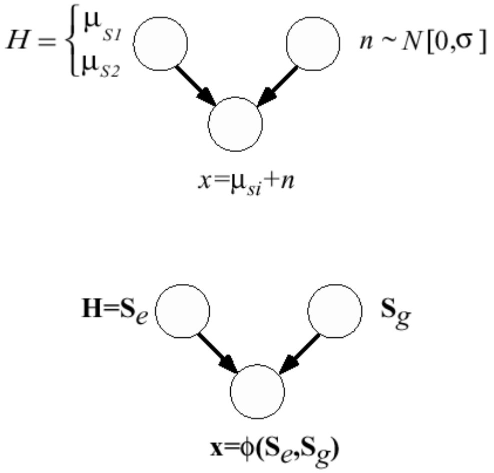
p(S1,S2,I)=p(I|S1,S2)p(S1)p(S2)

p(S,I1,I2)=p(I1|S)p(I2|S)p(S)

p(S,L,I)=p(I|L)p(L|S)p(S)

■ Primary, secondary variables.

The following figure draws a parallel between the causal structure, as determined by the generative model, for signal detection theory (as in the light detection problem), and the general problem of visual inference.


$H=\begin{cases}\mu_{S1}\\ \mu_{S2}\end{cases}$
$n \sim N[0,\sigma]$
$x=\mu_{si}+n$
$\mathbf{H=S}_e$
$\mathbf{S}_g$
$\mathbf{x=\phi(S}_e\mathbf{,S}_g\mathbf{)}$

We can interpret the causal structure in terms of conditional probability.

The top panel shows one possible generative graph structure for an ideal observer problem in classical signal detection theory (SDT). The data are determined by the signal hypotheses plus (additive gaussian) noise. Knowledge is represented by the joint probability p(x,H,n)=p(x|H,n)p(H)p(n). The lower panel shows a simplified example of the generative structure for perceptual inference from a pattern inference theory perspective. The image measurements (x) are determined by a typically non-linear function (\phi) of primary signal variables (S_e) and confounding secondary variables (S_g). Knowledge is represented by the joint probability p(x,S_e,S_g). Both scene and image variables can be high dimensional vectors. In general, the causal structure of natural image patterns is more complex and consequently requires elaboration of its graphical representation. For SDT and pattern inference theory, the task is to make a decision about the signal hypotheses or primary signal variables, while discounting the noise or secondary variables. Thus optimal perceptual decisions are determined by p(x,S_e), which is derived by summing over the secondary variables (i.e. marginalizing with respect to the secondary variables): $\int_{S\_g} p(x, S\_e, S\_g)\, d S\_g$.

Influences between variables are represented by conditioning, and a graphical model expresses the conditional independencies between variables. Two random variables may only become independent, however, once the value of some third variable is known. This is called conditional independence.Recall from above that two random variables are independent if and only if their joint probability is equal to the product of their individual probabilities. Thus, if p(A,B) = p(A)p(B), then A and B are independent. If p(A,B|C) = p(A|C)p(B|C), then A and B are conditionally independent. When corn prices drop in the summer, hay fever incidence goes up. However, if the joint on corn price and hay fever is conditioned on ``ideal weather for corn and ragweed", the correlation between corn prices and hay fever drops. This is because Corn price and hay fever symptoms are conditionally independent.

There is a correlation between eating ice cream and drowning. Why? What event should you condition on to make the dependence go away?

■ What is noise? Primary and secondary variables in SDT and in pattern inference theory

Noise is whatever you don't care to estimate, but contributes to the data.

Optimal Inference and task dependence: Fruit example

(due to James Coughlan; see Yuille, Coughlan, Kersten & Schrater).

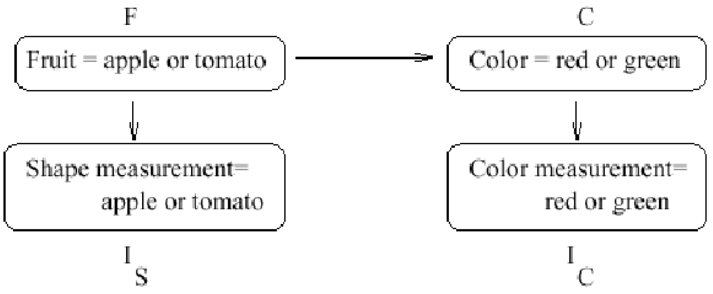

F
C
Fruit = apple or tomato
Color = red or green
Shape measurement= apple or tomato
Color measurement= red or green
I_S
I_C

Figure from Yuille, Coughlan, Kersten & Schrater.

The graph specifies how to decompose the joint probability:

p[F, C, Is, Ic] = p[Ic | C] p[C | F] p[Is | F] p[F]

Generative model: The prior model on hypotheses, F & C

More apples (F=1) than tomatoes (F=2), and:

```
In[286]:=  ppF[F_] := If[F == 1, 9 / 16, 7 / 16];
           TableForm[Table[ppF[F], {F, 1, 2}], TableHeadings -> {{"F=a", "F=t"}}]
```

```
Out[287]//TableForm=
F=a │ 9/16
F=t │ 7/16
```

The conditional probability **cpCF[C|F]**:

```
In[288]:=  cpCF[F_, C_] := Which[F == 1 && C == 1, 5 / 9, F == 1 && C == 2, 4 / 9,
              F == 2 && C == 1, 6 / 7, F == 2 && C == 2, 1 / 7];
           TableForm[Table[cpCF[F, C], {C, 1, 2}, {F, 1, 2}],
            TableHeadings -> {{"C=r", "C=g"}, {"F=a", "F=t"}}]
```

```
Out[289]//TableForm=
    │ F=a  F=t
────┼─────────
C=r │ 5/9  6/7
C=g │ 4/9  1/7
```

So the joint is:

```
In[290]:=  jpFC[F_, C_] := cpCF[F, C] ppF[F];
           TableForm[Table[jpFC[F, C], {F, 1, 2}, {C, 1, 2}],
            TableHeadings -> {{"F=a", "F=t"}, {"C=r", "C=g"}}]
```

```
Out[291]//TableForm=
    │ C=r   C=g
────┼──────────
F=a │ 5/16  1/4
F=t │ 3/8   1/16
```

We can marginalize to get the prior probability on color alone is:

In[292]:= ppC[C_] := $\sum_{F=1}^{2}$ jpFC[F, C]

Question: Is fruit identity independent of material color--i.e. is F independent of C? Check whether the joint probability on Fruit and Color can be factored into the product of the prior probabilities on Fruit and Color.

■ Answer

No.

In[293]:=

```
TableForm[Table[jpFC[F, C], {F, 1, 2}, {C, 1, 2}],
 TableHeadings -> {{"F=a", "F=t"}, {"C=r", "C=g"}}]
TableForm[Table[ppF[F] ppC[C], {F, 1, 2}, {C, 1, 2}],
 TableHeadings -> {{"F=a", "F=t"}, {"C=r", "C=g"}}]
```

Out[293]//TableForm=

```
    │ C=r   C=g
────┼───────────
F=a │ 5/16  1/4
    │
F=t │ 3/8   1/16
```

Out[294]//TableForm=

```
    │ C=r     C=g
────┼───────────────
F=a │ 99/256  45/256
    │
F=t │ 77/256  35/256
```

Generative model: The likelihood model--probabilities of measurements, i.e. some features given hypotheses

Suppose that we have gathered some "image statistics" which provides us knowledge of how the image measurements for shape Is, and for color Ic depend on the type of fruit F, and material color, C. For simplicity, our measurements are discrete and binary (a more realistic case, they would have continuous values), say Is = {am, tm}, and Ic = {rm, gm}.

P(I_S= am, tm | F=a) = {11/16, 5/16}

P(I_S= am, tm | F=t) = {5/8, 3/8}

P(I_C= rm, gm | C=r) = {9/16, 7/16}

P(I_C= rm, gm | C=g) = {1/2, 1/2}

We use the notation am, tm, rm, gm because the measurements are already suggestive of the likely cause. So there is a correlation between apple and apple-like shapes, am; and between red material, and "red" measurements. In general, there may not be an obvious correlation like this.

We define a function for the probability of Ic given C, **cpIcC[Ic | C]**:

In[295]:=

```
cpIcC[Ic_, C_] := Which[Ic == 1 && C == 1, 9 / 16, Ic == 1 && C == 2,
   7 / 16, Ic == 2 && C == 1, 1 / 2, Ic == 2 && C == 2, 1 / 2];
TableForm[Table[cpIcC[Ic, C], {C, 1, 2}, {Ic, 1, 2}],
 TableHeadings -> {{"Ic=rm", "Ic=gm"}, {"C=r", "C=g"}}]
```

Out[296]//TableForm=

```
      │C=r  C=g
──────┼─────────
      │ 9    1
Ic=rm │ ──   ─
      │ 16   2
      │ 7    1
Ic=gm │ ──   ─
      │ 16   2
```

The probability of Is conditional on F is **cpIsF[Is | F]**:

In[297]:=

```
cpIsF[Is_, F_] := Which[Is == 1 && F == 1, 11 / 16, Is == 1 && F == 2,
   5 / 8, Is == 2 && F == 1, 5 / 16, Is == 2 && F == 2, 3 / 8];
TableForm[Table[cpIsF[Is, F], {Is, 1, 2}, {F, 1, 2}],
 TableHeadings -> {{"Is=am", "Is=tm"}, {"F=a", "F=t"}}]
```

Out[298]//TableForm=

```
      │F=a  F=t
──────┼─────────
      │ 11   5
Is=am │ ──   ─
      │ 16   8
      │ 5    3
Is=tm │ ──   ─
      │ 16   8
```

The total joint probability

We now have enough information to put probabilities on the 2x2x2 "universe" of possibilities, i.e. all possible combinations of fruit, color, and image measurements. Looking at the graphical model makes it easy to use the product rule to construct the total joint, which is:

p[F, C, Is, Ic] = p[Ic | C] p[C | F] p[Is | F] p[F]:

In[299]:=

```
jpFCIsIc[F_, C_, Is_, Ic_ ] :=
 cpIcC[ Ic, C ] cpCF[F, C ] cpIsF[Is, F ] ppF[F ]
```

Usually, we don't need the probabilities of the image measurements (because once the measurements are made, they are fixed and we want to compare the probabilities of the hypotheses. But in our simple case here, once we have the joint, we can calculate the probabilities of the image measurements through marginalization p(Is,Ic)=$\sum_C \sum_F p(F, C, \text{Is}, \text{Ic})$, too:

In[300]:=

```
                     2   2
jpIsIc[Is_, Ic_] :=  ∑   ∑ jpFCIsIc[F, C, Is, Ic ]
                    C=1 F=1
```

Three MAP tasks

We are going to show that the best guess (i.e. maximum probability) depends on the task.

■ Define argmax[] function:

In[301]:=

```
argmax[x_] := Position[x, Max[x]];
```

■ Pick most probable fruit AND color--Answer "red tomato"

First, suppose the task is to make the best bet as to the fruit AND material color. To make it concrete, suppose that we see an "apple-ish shape" with a reddish color, i.e., we measure Is=am=1, and Ic = rm=1. The measurements suggest "red apple", but to find the most probable, we need to take into account the priors too in order to make the best guesses. **p(F,C | Is, Ic)** is given by:

In[302]:=

```
FCcIsIc[F_, C_, Is_, Ic_] := jpFCIsIc[F, C, Is, Ic ] / jpIsIc[Is, Ic]
```

In[303]:=

```
TableForm[FCcIsIcTable = Table[FCcIsIc[F, C, 1, 1 ], {F, 1, 2}, {C, 1, 2}],
  TableHeadings -> {{"F=a", "F=t"}, {"C=r", "C=g"}}]
Max[FCcIsIcTable]
argmax[FCcIsIcTable]
```

Out[303]//TableForm=

```
    | C=r    C=g
----+-------------
    |  55    308
F=a | ---   ----
    | 157   1413
    |  60     70
F=t | ---   ----
    | 157   1413
```

Out[304]=

```
 60
---
157
```

Out[305]=

```
{{2, 1}}
```

Note that we don't have to calculate the conditional. We can use the fact that p(F,C | Is, Ic) = $\frac{p(F,C,\mathrm{Is},\mathrm{Ic})}{p(\mathrm{Is},\mathrm{Ic})} \propto$ p(F, C , Is=1, Ic=1). I.e. the conditional is proportional to the function specified by the joint evaluated at Is=1, and Ic=1.

In[307]:=

```
TableForm[
 jpFCIsIcTable = Table[jpFCIsIc[F, C, 1, 1], {F, 1, 2}, {C, 1, 2}],
 TableHeadings -> {{"F=a", "F=t"}, {"C=r", "C=g"}}]
Max[jpFCIsIcTable]
argmax[jpFCIsIcTable]
```

Out[307]//TableForm=

```
    │ C=r    C=g
────┼────────────
    │ 495     77
F=a │ ────   ────
    │ 4096   1024
    │ 135     35
F=t │ ────   ────
    │ 1024   2048
```

Out[308]=

$$\frac{135}{1024}$$

Out[309]=

```
{{2, 1}}
```

■ Question: Is p(F, C, Is=1, Ic=1) a joint probability on F and C? (Answer: No.)

Redefine **jpFCcIsIcTable** and **jpFCIsIcTable**, without the TableForm[] wrapper and calculate:

In[332]:=

```
Total[jpFCcIsIcTable, {2}]
Total[jpFCIsIcTable, {2}]
```

In either case, we conclude that "Red tomato" is the most probable once we take into account the difference in priors.

■ Pick most probable color--Answer "red"

Same measurements as before. But now suppose we only care about the true material color, and not the identity of the object. Then we want to integrate out or marginalize with respect to the shape or fruit-type variable, F. In this case, we want to maximize the posterior:

p(C | Is=1, Ic=1)=$\sum_{F=1}^{2} p(F, C \mid \text{Is} = 1, \text{Ic} = 1)$

In[334]:=

```
pC[C_, Is_, Ic_] := Sum[jpFCcIsIc[F, C, Is, Ic], {F, 1, 2}]
```

$$\text{pC}[C\_, \text{Is}\_, \text{Ic}\_] := \sum_{F=1}^{2} \text{jpFCcIsIc}[F, C, \text{Is}, \text{Ic}]$$

Revise cell to work with *Mathematica* 7.

```
TableForm[pCTable = Table[pC[C, 1, 1], {C, 1, 2}],
   TableHeadings → {{"C=r", "C=g"}}];
pCTable = Table[pC[C,  1,  1 ], {C, 1, 2}]
Max[pCTable]
argmax[pCTable]
```

Answer is that the most probable material color is C = r, "red".

■ Pick most probable fruit--Answer "apple"

Same measurements as before. But now, we don't care about the material color, just the identity of the fruit. Then we want to integrate out or marginalize with respect to the material variable, C. In this case, we want to maximize the posterior:

p(F | Is, Ic)

In[316]:=

```
                     2
pF[F_, Is_, Ic_] := ∑ jpFCcIsIc[F,  C,  Is,  Ic ]
                    C=1
```

Revise cell to work with *Mathematica* 7.

In[317]:=

```
TableForm[pFTable = Table[pF[F,  1,  1 ], {F, 1, 2}],
  TableHeadings -> {{"F=a", "F=t"}}]
Max[pFTable]
argmax[pFTable]
```

The answer is "apple". So to sum up, for the same data measurements, the most probable fruit AND color is "red tomato", but the most probable fruit is "apple"!

■ Important "take-home message": *Optimal inference depends on the precise definition of the task*

Try expressing the consequences using the frequency interpretation of probability.

Bayesian learning of univariate Gaussian mean: MAP

From a statistical point of view, one form of learning is "density estimation" from histogram measurements. In high dimensions this is hard, but is easier if we have a low-dimensional parametric model for the density--i.e. the density is modeled in terms of a few parameters. So for example, the 1D Gaussian could be approximated by a huge list of numbers ("statistics")--one for each bin, each number is an estimate of the probability of the value of the random variable falling in that bin. But because it is Gaussian, we can be more efficient by representing the density in terms of just two numbers (also "statistics", but just the mean and variance), and a formula.

In this context, learning becomes *parameter estimation*.

A Bayesian learning example: Suppose we know the data comes from a Gaussian generative process, but we don't know the mean?

Suppose we have a set of samples that come from a Gaussian distribution with known variance σ^2, but unknown mean μ.

$$\begin{aligned} &x_i = \text{noise, where noise} \sim N[\mu, \sigma], \text{ or equivalently} \\ &x_i = \mu + \text{noise, where noise} \sim N[0, \sigma] \end{aligned} \tag{2}$$

```
In[277]:=  ndist0 = NormalDistribution[μ, σ];
```

Although we don't know the mean, we can assume a Gaussian prior on the mean:

$$\mu \sim N[\mu 0, \sigma 0] \tag{3}$$

```
In[278]:=  ndistμ = NormalDistribution[μ0, σ₀];
           PDF[ndistμ, μ]
```

Out[279]=

$$\frac{e^{-\frac{(\mu-\mu 0)^2}{2\sigma_0^2}}}{\sqrt{2\pi}\,\sigma_0}$$

I.e. we make an initial guess of the mean's mean (μ0) and standard deviation (σ0). But we are willing to change our estimate of the mean given new data. If we are really uncertain at the beginning,, we can start of with a large standard deviation, and as we gather data, the uncertainty about the value of the mean will decrease.

Suppose the generative model N[μ, σ] produces three i.i.d. (independent, identically distributed) samples x_1, x_2,x_3.What is the MAP estimate of μ? Which value of μ makes the posterior biggest? We use Bayes rule:

$$p(\mu \mid x_1, x_2, x_3) = \frac{p(x_1, x_2, x_3 \mid \mu)\, p(\mu)}{p(x_1, x_2, x_3)} \tag{4}$$

$p(x_1 \mid \mu)$ is given by :

```
PDF[ndist0, x₁]
```

$$\frac{e^{-\frac{(x_1-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

Because the samples are drawn independently, the $p(x_1, x_2, x_3 \mid \mu)$ is the product of three terms, so the numerator is $p(x_1 \mid \mu)p(x_2 \mid \mu)p(x_3 \mid \mu)$ times the prior $p(\mu)$:

```
In[280]:= PDF[ndist0, x1] * PDF[ndist0, x2] * PDF[ndist0, x3] * PDF[ndistμ, μ]
```

Out[280]=

$$\frac{e^{-\frac{(-\mu+x_1)^2}{2\sigma^2}-\frac{(-\mu+x_2)^2}{2\sigma^2}-\frac{(-\mu+x_3)^2}{2\sigma^2}-\frac{(\mu-\mu 0)^2}{2\sigma_0^2}}}{4\pi^2\sigma^3\sigma_0}$$

■ Calculating the MAP estimate of mean

To find the value of the mean that is largest given our three samples, and our prior assumption, we find μ where $p\,(x_1,\ x_2,\ x_3 \mid \mu)\ p\,(\mu)$ is biggest:

```
In[281]:= g = PDF[ndist0, x1] * PDF[ndist0, x2] * PDF[ndist0, x3] * PDF[ndistμ, μ];
          t = Log[g];
          t1 = PowerExpand[t]
          t2 = D[t1, μ]
          Solve[-t2 == 0, μ]
```

Out[283]=

$$-2\,\mathrm{Log}[2]-2\,\mathrm{Log}[\pi]-3\,\mathrm{Log}[\sigma]-\mathrm{Log}[\sigma_0]-\frac{(-\mu+x_1)^2}{2\,\sigma^2}-\frac{(-\mu+x_2)^2}{2\,\sigma^2}-\frac{(-\mu+x_3)^2}{2\,\sigma^2}-\frac{(\mu-\mu 0)^2}{2\,\sigma_0^2}$$

Out[284]=

$$\frac{-\mu+x_1}{\sigma^2}+\frac{-\mu+x_2}{\sigma^2}+\frac{-\mu+x_3}{\sigma^2}-\frac{\mu-\mu 0}{\sigma_0^2}$$

Out[285]=

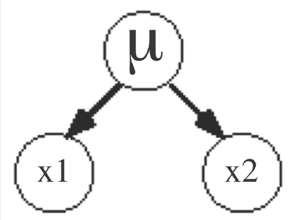
$$\left\{\left\{\mu\to\frac{\mu 0\,\sigma^2+x_1\,\sigma_0^2+x_2\,\sigma_0^2+x_3\,\sigma_0^2}{\sigma^2+3\,\sigma_0^2}\right\}\right\}$$

In general, one can update from n samples in batch mode:

$$\left\{\left\{\mu\to\frac{\frac{\mu 0}{\sigma 0^2}+\frac{1}{\sigma^2}\sum_{i=1}^{n}x_i}{\frac{n}{\sigma^2}+\frac{1}{\sigma 0^2}}\right\}\right\}\tag{5}$$

For the multi-variate case, see Duda and Hart.

Using a similar derivation to that above, find the optimal rule for integrating two measurements to estimate the mean. Assume the Gaussian case, with conditional independence (as represented in graph below):



Answer:

$$\mu = \frac{r_1}{r_1 + r_2} x_1 + \frac{r_2}{r_1 + r_2} x_2$$

where $r_i = 1/\sigma_i^2$. Ma et al. (2006) showed conditions under which how neural populations could achieve optimal estimates by summing spikes.

What is the influence of the initial estimate of the mean as learning goes on? What is the estimate of the mean as n gets large?

Appendices

Using *Mathematica* lists to manipulate discrete priors, likelihoods, and posteriors

■ A note on list arithmetic

We haven't done standard matrix/vector operations above to do conditioning. We've take advantage of how *Mathematica* divides a 2x3 array by a 2-element vector:

```
M=Array[m,{2,3}]
X = Array[x,{2}]
```

$$\begin{pmatrix} m(1,1) & m(1,2) & m(1,3) \\ m(2,1) & m(2,2) & m(2,3) \end{pmatrix}$$

$\{x(1), x(2)\}$

```
M/X
```

$$\begin{pmatrix} \frac{m(1,1)}{x(1)} & \frac{m(1,2)}{x(1)} & \frac{m(1,3)}{x(1)} \\ \frac{m(2,1)}{x(2)} & \frac{m(2,2)}{x(2)} & \frac{m(2,3)}{x(2)} \end{pmatrix}$$

■ Putting the probabilities back together again to get the joint

```
Transpose[Transpose[pHx] px]
```

$\left(\text{px}\,\text{pHx}^{\mathsf{T}}\right)^{\mathsf{T}}$

```
pxH pH
```

pH pxH

■ Getting the posterior from the priors and likelihoods:

One reason Bayes' theorem is so useful is that it is often easier to formulate the likelihoods (e.g. from a causal or generative model of how the data could have occurred), and the priors (often from heuristics, or in computational vision empirically testable models of the external visual world). So let's use *Mathematica* to derive **p(H|x)** from **p(x|H**) and **p(H)** , (i.e. pHx from pxH and pH).

```
px2 = Plus @@ (pxH pH)
```

pH + pxH

```
Transpose[Transpose[(pxH pH)] / Plus @@ (pxH pH)]
```

$$\frac{(\text{pH}\,\text{pxH})^{\mathsf{T}\,\mathsf{T}}}{\text{pH}+\text{pxH}}$$

■ Show that this joint probability has a uniform prior (i.e. both priors equal).

```
p = {{1 / 8, 1 / 8, 1 / 4}, {1 / 4, 1 / 8, 1 / 8}}
```

$$\begin{pmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{pmatrix}$$

Marginalization and conditioning: A small dimensional example using list manipulation in *Mathematica*

■ A discrete joint probability

All of our knowledge regarding the signal discrimination problem can be described in terms of the joint probability of the hypotheses, **H** and the possible data measurements, **x.** The probability function assigns a number to all possible combinations:

p[H, x]

That is, we are assuming that both the hypotheses and the data are discrete random variables.

$$\mathbf{H} = \begin{cases} \mathbf{S1} \\ \mathbf{S2} \end{cases}$$

```
x ∈ {1, 2, ...}
```

Let's assume that x can only take on one of three values, 1, 2, or 3. And suppose the joint probability is:

$$\mathbf{p} = \left\{\left\{\frac{1}{12}, \frac{1}{12}, \frac{1}{6}\right\}, \left\{\frac{1}{3}, \frac{1}{6}, \frac{1}{6}\right\}\right\}$$

$$\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}$$

```
TableForm[p, TableHeadings -> {{"H=S1", "H=S2"}, {"x=1", "x=2", "x=3"}}]
```

```
     | x=1   x=2   x=3
-----+----------------
H=S1 | 1/12  1/12  1/6
H=S2 | 1/3   1/6   1/6
```

The total probability should sum up to one. Let's test to make sure. We first turn the list of lists into a singel list of scalars using **Flatten[]**. And then we can sum either with **Apply[Plus,Flatten[p]].**

```
Plus @@ Flatten[p]
```

```
1
```

We can pull out the first row of p like this:

```
p[[1]]
```

$$\left\{\frac{1}{12}, \frac{1}{12}, \frac{1}{6}\right\}$$

$$\boldsymbol{\left\{\frac{1}{12}, \frac{1}{12}, \frac{1}{6}\right\}}$$

$$\left\{\frac{1}{12}, \frac{1}{12}, \frac{1}{6}\right\}$$

Is this the probability of x? No. For a start, the numbers don't sum to one. But we can get it through the two processes of marginalization and conditioning.

■ Marginalizing

What are the probabilities of the data, p(x)? To find out, we use the *sum rule* to sum over the columns:

```
px = Apply[Plus, p]
```

$$\left\{\frac{5}{12}, \frac{1}{4}, \frac{1}{3}\right\}$$

"Summing over "is also called **marginalization** or **"integrating out".** Note that marginalization turns a probability function with higher degrees of freedom into one of lower degrees of freedom.

What are the prior probabilities? p(H)? To find out, we sum over the rows:

```
pH = Apply[Plus, Transpose[p]]
```

$$\left\{\frac{1}{3}, \frac{2}{3}\right\}$$

Conditioning

Now that we have the marginals, we can get use the *product rule* to obtain the conditional probability through conditioning of the joint:

$$p[x \mid H] = \frac{p[H,\ x]}{p[H]}$$

Set::write : Tag List in $\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[x \mid H]$ is Protected. ≫

$$\frac{\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[H, x]}{\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[H]}$$

In the Exercises, you can see how to use *Mathematica* to do the division for conditioning. The syntax is simple:

```
pxH = p / pH
```

$$\begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}$$

Note that the probability of x conditional on H sums up to 1 over x, i.e. each row adds up to 1. But, the columns do not. **p[xlH]** is a **probability** function of x, but a **likelihood** function of H. The posterior probability is obtained by conditioning on x:

$$p[H \mid x] = \frac{p[H,\ x]}{p[x]}$$

Set::write : Tag List in $\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[H \mid x]$ is Protected. ≫

$$\frac{\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[H, x]}{\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[x]}$$

Syntax here is a bit more complicated, because the number of columns of px don't match the number of rows of p. We use Transpose[] to exchange the columns and rows of p before dividing, and then use Transpose again to get back the 2x3 form:

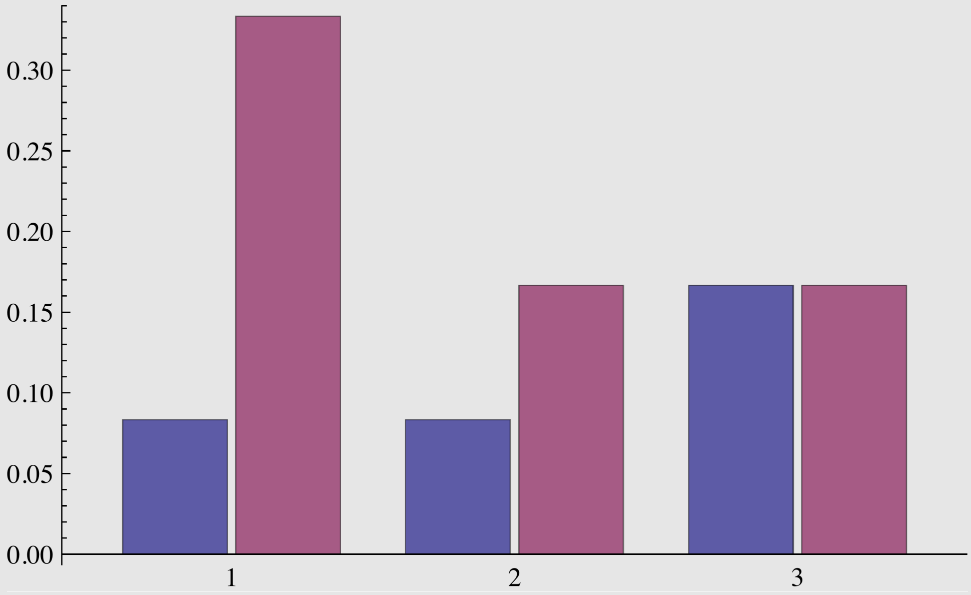
```
pHx = Transpose[Transpose[p] / px]
```

$$\begin{pmatrix} \frac{1}{5} & \frac{1}{3} & \frac{1}{2} \\ \frac{4}{5} & \frac{2}{3} & \frac{1}{2} \end{pmatrix}$$

Plotting the joint

The following BarChart[] graphics function requires in add-in package (**<< Graphics`Graphics`**), which is specified at the top of the notebook. You could also use **ListDensityPlot[]**.

```
BarChart[p⟦1⟧, p⟦2⟧]
```



Marginalization and conditioning: An example using *Mathematica* functions

A discrete joint probability

All of our knowledge regarding the signal discrimination problem can be described in terms of the joint probability of the hypotheses, **H** and the possible data measurements, **x.** The probability function assigns a number to all possible combinations:

p[H, x]

That is, we are assuming that both the hypotheses and the data are discrete random variables.

```
H = ⎧ S1
    ⎩ S2

x ∈ {1, 2, ...}
```

Let's assume that x can only take on one of three values, 1, 2, or 3. And suppose the joint probability is:

```
p[H_, x_] := Which[H == 1 && x == 1, 1 / 12, H == 1 && x == 2, 1 / 12,
   H == 1 && x == 3, 1 / 6, H == 2 && x == 1, 1 / 3, H == 2 && x == 2, 1 / 6,
   H == 2 && x == 3, 1 / 6];
```

SetDelayed::write : Tag List in $\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}$[H_, x_] is Protected. ≫

```
TableForm[Table[p[H, x], {H, 1, 2}, {x, 1, 3}],
 TableHeadings -> {{"H=s1", "H=s2"}, {"X=1", "X=2", "X=3"}}]
```

$$\begin{array}{l|lll}
 & \text{X=1} & \text{X=2} & \text{X=3} \\
\hline
\text{H=s1} & \begin{matrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{matrix}[1, 1] & \begin{matrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{matrix}[1, 2] & \begin{matrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{matrix}[1, 3] \\
\text{H=s2} & \begin{matrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{matrix}[2, 1] & \begin{matrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{matrix}[2, 2] & \begin{matrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{matrix}[2, 3]
\end{array}$$

The total probability should sum up to one. Let's test to make sure. We first turn the list of lists into a singel list of scalars using **Flatten[]**. And then we can sum either with **Apply[Plus,Flatten[p]].**

```
Sum[p[H, x], {H, 1, 2}, {x, 1, 3}]
```

$$\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[1, 1] + \begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[1, 2] + \begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[1, 3] +$$
$$\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[2, 1] + \begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[2, 2] + \begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[2, 3]$$

We can pull out the first row of p like this:

```
Table[p[1, x], {x, 1, 3}]
```

$$\left\{\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[1, 1], \begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[1, 2], \begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[1, 3]\right\}$$

Is this the probability of x? No. For a start, the numbers don't sum to one. But we can get it through the two processes of marginalization and conditioning.

Marginalizing

What are the probabilities of the data, p(x)? To find out, we use the *sum rule* to sum over the columns:

```
px[x_] := Sum[p[H, x], {H, 1, 2}];
```

SetDelayed::write : Tag List in $\left\{\frac{5}{12}, \frac{1}{4}, \frac{1}{3}\right\}$[x_] is Protected. ≫

```
Table[px[x], {x, 1, 3}]
```

$$\left\{\left\{\frac{5}{12}, \frac{1}{4}, \frac{1}{3}\right\}[1], \left\{\frac{5}{12}, \frac{1}{4}, \frac{1}{3}\right\}[2], \left\{\frac{5}{12}, \frac{1}{4}, \frac{1}{3}\right\}[3]\right\}$$

"Summing over "is also called **marginalization** or **"integrating out".** Note that marginalization turns a probability function with higher degrees of freedom into one of lower degrees of freedom.

What are the prior probabilities? p(H)? To find out, we sum over the rows:

```
pH[H_] := Sum[p[H, x], {x, 1, 3}];
```

SetDelayed::write : Tag List in $\left\{\frac{1}{3}, \frac{2}{3}\right\}$[H_] is Protected. ≫

```
Table[pH[H], {H, 1, 2}]
```

$$\left\{\left\{\frac{1}{3}, \frac{2}{3}\right\}[1], \left\{\frac{1}{3}, \frac{2}{3}\right\}[2]\right\}$$

■ Conditioning

Now that we have the marginals, we can get use the *product rule* to obtain the conditional probability through conditioning of the joint:

$$p[x\,|\,H]=\frac{p[H,\;x]}{p[H]}$$

Set::write : Tag List in $\begin{pmatrix}\frac{1}{12}&\frac{1}{12}&\frac{1}{6}\\ \frac{1}{3}&\frac{1}{6}&\frac{1}{6}\end{pmatrix}[x\,|\,H]$ is Protected. ≫

$$\frac{\begin{pmatrix}\frac{1}{12}&\frac{1}{12}&\frac{1}{6}\\ \frac{1}{3}&\frac{1}{6}&\frac{1}{6}\end{pmatrix}[H,x]}{\begin{pmatrix}\frac{1}{12}&\frac{1}{12}&\frac{1}{6}\\ \frac{1}{3}&\frac{1}{6}&\frac{1}{6}\end{pmatrix}[H]}$$

We use function definition in *Mathematica* to do the division for conditioning. The syntax is simple:

```
pxH[H_, x_] := p[H, x] / pH[H];
```

SetDelayed::write : Tag List in $\begin{pmatrix}\frac{1}{4}&\frac{1}{4}&\frac{1}{2}\\ \frac{1}{2}&\frac{1}{4}&\frac{1}{4}\end{pmatrix}$[H_, x_] is Protected. ≫

```
Table[pxH[H, x], {H, 1, 2}, {x, 1, 3}]
```

$$\begin{pmatrix}\begin{pmatrix}\frac{1}{4}&\frac{1}{4}&\frac{1}{2}\\ \frac{1}{2}&\frac{1}{4}&\frac{1}{4}\end{pmatrix}[1,1] & \begin{pmatrix}\frac{1}{4}&\frac{1}{4}&\frac{1}{2}\\ \frac{1}{2}&\frac{1}{4}&\frac{1}{4}\end{pmatrix}[1,2] & \begin{pmatrix}\frac{1}{4}&\frac{1}{4}&\frac{1}{2}\\ \frac{1}{2}&\frac{1}{4}&\frac{1}{4}\end{pmatrix}[1,3]\\ \begin{pmatrix}\frac{1}{4}&\frac{1}{4}&\frac{1}{2}\\ \frac{1}{2}&\frac{1}{4}&\frac{1}{4}\end{pmatrix}[2,1] & \begin{pmatrix}\frac{1}{4}&\frac{1}{4}&\frac{1}{2}\\ \frac{1}{2}&\frac{1}{4}&\frac{1}{4}\end{pmatrix}[2,2] & \begin{pmatrix}\frac{1}{4}&\frac{1}{4}&\frac{1}{2}\\ \frac{1}{2}&\frac{1}{4}&\frac{1}{4}\end{pmatrix}[2,3]\end{pmatrix}$$

Note that the probability of x conditional on H sums up to 1 over x, i.e. each row adds up to 1. But, the columns do not. **p[xlH]** is a **probability** function of x, but a **likelihood** function of H. The posterior probability is obtained by conditioning on x:

$$p[\,H \mid x] = \frac{p[H,\ x]}{p[x]}$$

Set::write : Tag List in $\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[H \mid x]$ is Protected. ≫

$$\frac{\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[H, x]}{\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}[x]}$$

```
pHx[H_, x_] := p[H, x] / px[x];
```

SetDelayed::write : Tag List in $\begin{pmatrix} \frac{1}{5} & \frac{1}{3} & \frac{1}{2} \\ \frac{4}{5} & \frac{2}{3} & \frac{1}{2} \end{pmatrix}$[H_, x_] is Protected. ≫

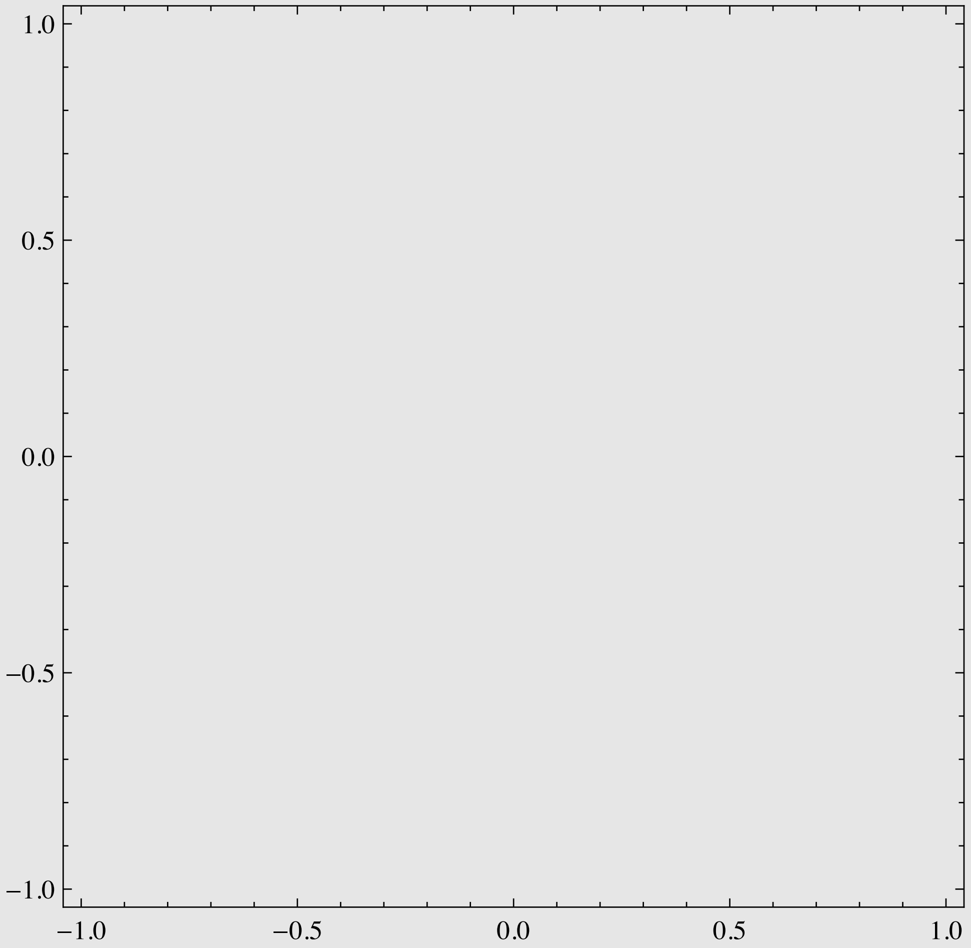
```
Table[pHx [H, x], {H, 1, 2}, {x, 1, 3}]
```

$$\begin{pmatrix} \begin{pmatrix} \frac{1}{5} & \frac{1}{3} & \frac{1}{2} \\ \frac{4}{5} & \frac{2}{3} & \frac{1}{2} \end{pmatrix}[1, 1] & \begin{pmatrix} \frac{1}{5} & \frac{1}{3} & \frac{1}{2} \\ \frac{4}{5} & \frac{2}{3} & \frac{1}{2} \end{pmatrix}[1, 2] & \begin{pmatrix} \frac{1}{5} & \frac{1}{3} & \frac{1}{2} \\ \frac{4}{5} & \frac{2}{3} & \frac{1}{2} \end{pmatrix}[1, 3] \\ \begin{pmatrix} \frac{1}{5} & \frac{1}{3} & \frac{1}{2} \\ \frac{4}{5} & \frac{2}{3} & \frac{1}{2} \end{pmatrix}[2, 1] & \begin{pmatrix} \frac{1}{5} & \frac{1}{3} & \frac{1}{2} \\ \frac{4}{5} & \frac{2}{3} & \frac{1}{2} \end{pmatrix}[2, 2] & \begin{pmatrix} \frac{1}{5} & \frac{1}{3} & \frac{1}{2} \\ \frac{4}{5} & \frac{2}{3} & \frac{1}{2} \end{pmatrix}[2, 3] \end{pmatrix}$$

Plotting the joint

We use **ListDensityPlot[]**.

```
ListDensityPlot[Table[p[H, x], {H, 1, 2}, {x, 1, 3}]]
```



■ Random number generator, a non-Gaussian example: The von Mises distribution, with Matlab code

(courtesy, Paul Schrater)

```
function pofx = vonMisespdf(x,mu,sigma)

% For -pi <= x <= pi

% force x-mu within -pi to pi

y = angle(exp(i*(x-mu)));

kappa = 1/(sigma)^2;

%kappa = sigma;

pofx = exp(kappa*cos(y))/(2*pi*besseli(0,kappa));

function vonrand = vonMisesrand(nrand,mu,sigma)

% inverse cumulative method, executed by table lookup with

% linear interpolation

% build sampled cdf
```

```
x = (-pi:2*pi/(2e3):pi);

pofx = vonMisespdf(x,0,sigma);

cofx = cumsum(pofx/sum(pofx));

u = rand(1,nrand);

vonrand = interp1(cofx,x,u)+mu;
```

References

Applebaum, D. (1996). Probability and Information . Cambridge, UK: Cambridge University Press.

Cover, T. M., & Joy, A. T. (1991). *Elements of Information Theory*. New York: John Wiley & Sons, Inc.

Duda, R. O., & Hart, P. E. (1973). Pattern classification and scene analysis . New York.: John Wiley & Sons.

Golden, R. (1988). A unified framework for connectionist systems. Biological Cybernetics, 59, 109-120.

Intrator, N. Combining Exploratory Projection Pursuit and Projection Pursuit Regression. Neural Computation (5):443-455, 1993. http://www.physics.brown.edu/users/faculty/intrator/papers/epp-ppr.ps.gz

Kersten, D. and P.W. Schrater (2000), *Pattern Inference Theory: A Probabilistic Approach to Vision*, in *Perception and the Physical World*, R. Mausfeld and D. Heyer, Editors. , John Wiley & Sons, Ltd.: Chichester. (pdf)

Kersten, D., Mamassian P & Yuille A (in press) Object perception as Bayesian inference. Annual Review of Psychology. (pdf, http://arjournals.annualreviews.org/doi/pdf/10.1146/annurev.psych.55.090902.142005)

Kersten, D., & Yuille, A. (2003). Bayesian models of object perception. Current Opinion in Neurobiology, 13(2) http://gandalf.psych.umn.edu/~kersten/kersten-lab/papers/KerstenYuilleApr2003.pdf

Ma, W. J., Beck, J. M., Latham, P. E., & Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nat Neurosci, 9*(11), 1432-1438.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge, UK: Cambridge University Press.

Yuille, A., Coughlan J., Kersten D.(1998) (pdf)