Introduction to Neural Networks
U. Minn. Psy 5038

More probability

■ Initialize standard library files:

In[1]:=
```
Off[General::spell1];
```

The next package is needed for the add-on multivariate gaussian

In[2]:=
```
<< Statistics`MultinormalDistribution`
```

Goals

Review the basics of probability distributions and statistics

More on generative modeling: drawing samples

Graphical models for inference

Optimal inference and Task dependence

Probability overview

Random variables, discrete probabilities, probability densities, cumulative distributions

■ Discrete: random variable X can take on a finite set of discrete values

X = {x(1),...,x(N)]

$$\sum_{i=1}^{N} p_i = \sum_{i=1}^{N} p(X = x(i)) = 1$$

■ Densities: X takes on continuous values, x, in some range.

Density : $p(x)$

Analogous to material mass,
we can think of the probability over some small domain of the random variable as " probability mass " :

$$\text{prob}(x < X < \text{dx} + x) = \int_{X}^{\text{dX}+X} p(x)\, dx$$

$$\text{prob}(x < X < \text{dx} + x) \simeq p(x)\,\text{dx}$$

Crudely speaking, however, an object (event space) always weighs 1 :

$$\int_{-\infty}^{\infty} p(x)\, dx = 1$$

Cumulative distribution:

$$\text{prob}\,(X < x) = \int_{-\infty}^{x} p(X)\, dX$$

■ Densities of discrete random variables

The Dirac Delta function, δ[•], allows us to use the mathematics of continuous distributions for discrete ones, by defining the density as:

$$\text{p[x]}=\sum_{i=1}^{N} p_i\, \delta[\text{x - x[i]}], \text{ where } \delta[\text{x - x[i]}] = \begin{cases} \infty & \text{for } x = x[i] \\ 0 & \text{for } x \neq x[i] \end{cases}$$

Think of the delta function, δ[•], as ϵ wide and 1/ϵ tall, and then let ϵ -> 0, so that:

$$\int_{-\infty}^{\infty} \delta(y)\, dy = 1$$

The density, p[x], is a series of spikes. It is infinitely high only at those points for which x = x[i], and zero elsewhere. But "infinity" is scaled so that the local mass or area around each point x[i], is p_i .

■ Joint probabilities

Prob $(X \text{ AND } Y) = p(X, Y)$
Joint density : $p(x, y)$

Three basic rules of probability

Suppose we know everything there is to know about a set of variables (A,B,C,D,E). What does this mean in terms of probability? It means that we know the joint distribution, p(A,B,C,D,E). In other words, for any particular combination of values (A=a,B=b, C=c, D=d,E=e), we can calculate, look up in a table, or determine some way or another the number p(A=a,B=b, C=c, D=d,E=e).

Deterministic relatinships are special cases. For example, suppose we know that there are only two specific pairs of numbers that determine Y as a function of X: {y1 = 2x1,y2=2x2,y3=2x3}, exactly. Then p(x1,y1)=

Rule 1: Conditional probabilities from joints: The product rule

Probability about an event changes when new information is gained.

Prob(X given Y) = p(X|Y)

$$p(X \mid Y) = \frac{p(X, Y)}{p(Y)}$$

$$p(X, Y) = p(X \mid Y)\, p(Y)$$

The form of the product rule is the same for densities as for probabilities.

Rule 2: Lower dimensional probabilities from joints: The sum rule (marginalization)

$$p(X) = \sum_{i=1}^{N} p(X, Y(i))$$

$$p(x) = \int_{-\infty}^{\infty} p(x, y)\, dx$$

Rule 3: Bayes' rule

From the product rule, and since p[X,Y] = p[Y,X], we have:

$p(Y \mid X) = \frac{p(X \mid Y)\, p(Y)}{p(X)}$, and using the sum rule, $p(Y \mid X) = \frac{p(X \mid Y)\, p(Y)}{\sum_Y p(X, Y)}$

Bayes Terminology in inference

Suppose we have some partial data (see half of someone's face), and we want to recall or complete the whole. Or suppose that we hear a voice, and from that visualize the face. These are both problems of statistical inference. We've already studied how

We typically think of the **Y** term as a random variable over the hypothesis space (a face), and **X** as data or a stimulus (partiall face, or sound). So for recalling a pattern **Y** from an input stimulus **X,** We'd like to have a function that tells us:

p(Y | X) which is called the **posterior** probability of the hypothesis (face) given the stimulus (partial face or sound).

-- i.e. what you get when you condition the joint by the stimulus data. The posterior is often what we'd like to base our decisions on, because it can be proved that picking the hypothesis **Y** which maximizes the posterior (i.e. maximum a posteriori or **MAP** estimation) minimizes the average probability of error.

p(Y) is the **prior** probability of the hypothesis (e.g. Given a context, such as your room, some faces are a priori more likely than others. For me an image patch stimulating my retina in my kitchen is much more likely to be my wife's than my brother's (who lives in another state)).

p(X|Y) is the **likelihood** of the hypothesis. Note this is a probability of **X**, but not of **Y**.(The sum over X is one, but the sum over Y isn't necessarily one.)

Independence

Knowledge of one event doesn't change the probability of another event.

p(X)=p(X|Y)

p(X,Y)=p(X)p(Y)

Density mapping theorem

Suppose we have a change of variables that maps a discrete set of x's uniquely to y's: X->Y.

Discrete random variables

No change to probability function. The mapping just corresponds to a change of labels, so the probabilities p(X)=p(Y).

Continuous random variables

Form of probability density function does change because we require the probability "mass" to be unchanged: p(x)dx = p(y)dy

Suppose, y=f(x)

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

$$p_Y(\mathrm{y}) = \int \delta(y - f(x))\, f^{-1}(x)\, p_X(x)\, dx$$

over each monotonic part of f.

Convolution theorem for adding rvs

Let x be distributed as g(x), and y as h(x). Then the probability density for z=x+y is, f(z):

$$f(z) = \int g(s)\, h(z - s)\, ds \tag{1}$$

Statistics

■ Expectation & variance

Analogous to center of mass:

Definition of expectation or average:

$\mathrm{Average}[X] = \bar{X} = E[X] = \Sigma\, x[i]\, p[x[i]]$

$$\mu = E[X] = \int x\, p(x)\, \mathrm{dx} \sim \sum_{i=1}^{N} x_i / N$$

Some rules:

E[X+Y]=E[X]+E[Y]

E[aX]=aE[X]

E[X+a]=a+E[X]

Definition of variance:

$\sigma^2 = \mathrm{Var}[X] = \mathrm{E}[[X\text{-}\mu]\text{^}2] = \sum_{j=1}^{N} ((p(x(j)))\,(x(j) - \mu))^2 = \sum_{j=1}^{N} p_j(x_j - \mu))^2$

$$\mathrm{Var}[X] = \int (x - \mu)^2\, p(x)\, \mathrm{dx} \sim \sum_{i=1}^{N} (x_i - \mu)^2 / N$$

Standard deviation:

$\sigma = \sqrt{\mathrm{Var}[X]}$

Some rules:

$\mathrm{Var}[X] = E[X^2] - E[X]^2$
$\mathrm{Var}[aX] = a^2\, \mathrm{Var}[X]$

■ Covariance & Correlation

Covariance:

Cov[X,Y] =E[[X - μ_X] [Y - μ_Y]]

Correlation coefficient:

$$\rho[X, Y] = \frac{\mathrm{Cov}[X, Y]}{\sigma_X\, \sigma_Y}$$

■ Cross and Autocovariance matrix

Suppose X and Y are vectors: $\{X_1, X_2, ...\}$ and $\{Y_1, Y_2, ...\}$

Cov[X_i ,Y_j] =E[[X_i - μ_{X_i}] [Y_j - μ_{Y_j}]] ~ $\sum_{n=1}^{N} (x_i^n - \mu_{X_i})\left(y_j^n - \mu_{Y_j}\right)^T / N$

Autocov[X_i ,X_j] =E[[X_i - μ_{X_i}] [X_j - μ_{Y_j}]] ~ $\sum_{n=1}^{N} (x_i^n - \mu_{X_i})\,(x_j^n - \mu_{X_i})^T / N$

In other words, the autocovariance matrix can be approximated by the outer product. It is a Hebbian matrix memory of pair-wise relationships.

■ Independent random variables

If p(X,Y)=p(X)p(Y), then

$E[X\, Y] = E[X]\, E[Y]$ (uncorrelated)
$\mathrm{Cov}[X, Y] = \rho[X, Y] = 0$
$\mathrm{Var}[X + Y] = \mathrm{Var}[X] + \mathrm{Var}[Y]$

If two random variables are uncorrelated, they are not necessarily independent.

Two random variables are said to be orthogonal if their correlation is zero.

Degree of belief vs., relative frequency

What is the probability that the Vikings will win the Superbowl in 2002? Assigning a number between 0 and 1 is assigning a degree of belief. These probabilities are also called subjective probabilities. "Odds" determine subjective probabilities, where the "odds of **x** to **y**" means probability **= x/(x+y)**.

What is the probability that a coin will come up heads? In this case, we can do an experiment. Flip the coin n times, and count the number of heads, say h[n], and then set the probability, p = h[n]/n -- the relative frequency . Of course, if we did it again, we may not get the same estimate of p. One solution often given is:

$$p = \lim_{n \to \infty} \frac{h(n)}{n}$$

A problem with this, is that there is no guarantee that a well − defined limit exists.

In some domains we can measure statistics, and model probabilities of both inputs and outputs. So the relative frequency interpretation seems reasonable. In practice, the dimensions of many problems in perception, cognition, and memory are so high, that it is impractical to do this. Once we use the statistical framework to model perception, say of a particular cue (say), then probabilities are more like "subjective unconscious beliefs".

Principle of insufficient reason

■ Principle of symmetry

Suppose we have N events, x[1],x[2],x[3],...,x[N] that are all physically identical except for the label. Then assume that

$$\text{prob}(x(1)) = \text{prob}(x(2)) = \text{prob}(x(3)) = \text{prob}(x(N)) = \frac{1}{N}$$

In other words,if we have no additional information about the events,we should assume that they are uniformly distributed. I.e., assume a uniform prior.

What about the continous case where there is no reason to assume any particular value at all between -∞ and +∞?

Improper priors.

■ Information theory and Maximum entropy

Information theory provides a powerful extension to the principle of symmetry. Information of event X is:

$$\text{Information}[X] = -\log_2(p(X))$$

Using the definition of expectation above, we can specify the expectation of information, which is called entropy. Entropy of a random variable X with probability distribution p[X] is:

$$H(X) = \text{Average}(\text{Information}[X]) = -\sum_X p(X)\log_2(p(X))$$

It can be shown that out of all possible probability distributions, H(X) is biggest for the uniform distribution, p(X)=1/N. Maximum entropy is looking like symmetry.

It turns out that a more powerful formulation of the principle of symmetry is maximum entropy. For example, out of all possible probability distributions of a random variable with infinity range, but with a specific mean and standard deviation, the Gaussian is unique in having the largest entropy. If the range goes from zero to infinity, and we know the mean, the maximum entropy distribution is an exponential (Cover and Thomas).

An interesting application of the maximum entropy principle is to learning image textures joint probabilities: p(I[1],...,I[N]), where N is very big, but where one has only a relatively small number of measured statistics relative to the number of possible images (which is really huge). The measurements underdetermine the dimesionality of the probability space--i.e. there are many different probability distributions which give the same statistics. So the principle of symmetry, or insufficient reason, says to choose the one with the maximum entropy.

More on generative modeling: Multivariate gaussian, mixtures

Making a univariate (scalar) gaussian random number generator

We'll use the density mapping theorem to turn uniformly distributed random numbers **Random[]** into gaussian distributed random numbers with mean =0 and standard deviation =1.

■ The Gaussian distribution

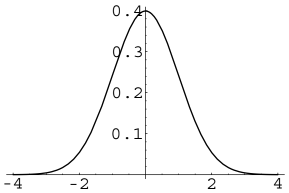
In[5]:=
```
Integrate[Exp[-(x - x0)^2 / (2 * σ^2) ], {x, -Infinity, Infinity}]
```

Out[5]=

$$\text{If}\left[\text{Re}[\sigma^2] > 0,\ \sqrt{2\pi}\ \sqrt{\sigma^2},\ \int_{-\infty}^{\infty} e^{-\frac{(x-x0)^2}{2\sigma^2}}\, dx\right]$$

Let x0=0 and σ=1:

In[6]:=
```
Plot[Exp[-(x1^2 ) / 2] / (Sqrt[2 * Pi]), {x1, -4, 4}];
```



Plot[PDF[NormalDistribution[0,1],x1],{x1,-4,4}]; gives the same thing using the add-on normal distribution function.

■ Cumulative gaussian

In[7]:=
```
Clear[cumulgauss, x, x1];
cumulgauss[x_] :=
 Integrate[Exp[-(x1^2 ) / 2] / (Sqrt[2 * Pi]), {x1, -Infinity, x}]
```

In[9]:=

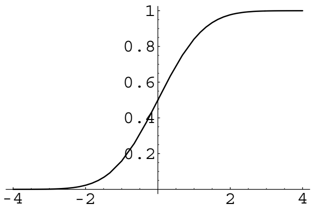
```
cumulgauss[Infinity]
```

Out[9]=

```
1
```

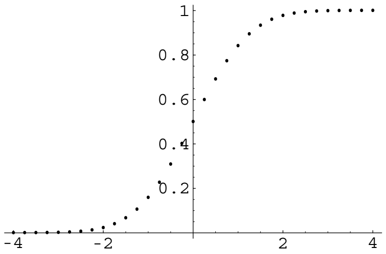
In[10]:=

```
Plot[cumulgauss[x], {x, -4, 4}];
```



In[11]:=

```
lcumulgauss = Table[{x, cumulgauss[x]}, {x, -4.0, 4.0, .25}];
ListPlot[lcumulgauss];
```



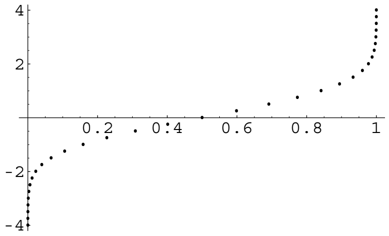
■ Make inverse cumulative gaussian table

In[13]:=

```
invlcumulgauss = RotateLeft[lcumulgauss, {1, 1}];
```

In[14]:=

```
ListPlot[invlcumulgauss];
```



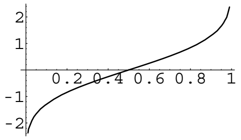
■ Make interpolated function of the inverse cumulative

In[15]:=

```
interinvlcumulgauss = Interpolation[invlcumulgauss];
```

In[16]:=

```
Plot[interinvlcumulgauss[x], {x, .01, .99}];
```



■ Draw samples with a standard deviation of Sqrt[10]

In[17]:=

```
Round[10 * interinvlcumulgauss[Random[]]]
```

Out[17]=

```
-1
```

■ Draw a bunch of samples, and plot up histogram

In[18]:=

```
z = Table[Round[10 * interinvlcumulgauss[Random[]]], {10000}];
domain = Range[-20, 20];
Freq = Map[Count[z, #] &, domain];
```

In[21]:=

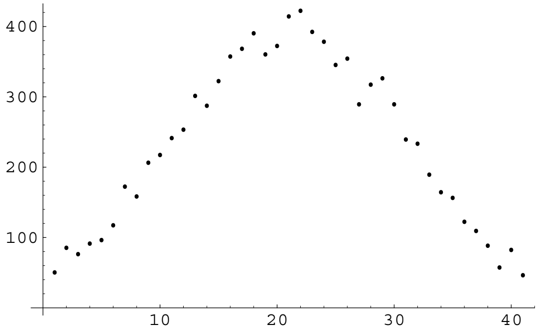
```
Sqrt[10.]
```

Out[21]=

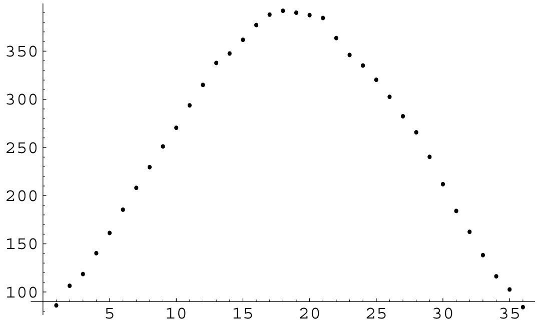
```
3.16228
```

In[22]:=

```
ListPlot[Freq];
```

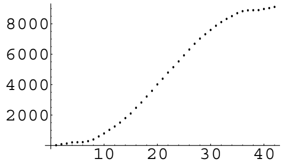


```
<< Statistics`DataSmoothing`
ListPlot[MovingAverage[Freq, 6]];
```



■ Plot up cumulative histogram

```
CumFreq = FoldList[Plus, 0, Freq];
ListPlot[CumFreq];
```



```
Transpose[{domain, Freq, CumFreq}] // MatrixForm
```

Multivariate (vector) gaussian distributions

■ Define multivariate gaussian probability density

An n-variate multivariate gaussian (multinormal) distribution with mean vector μ and covariance matrix Σ is denoted $N_n(\mu, \Sigma)$. The density is:

$$p(x) = \frac{1}{(2\pi)^{n/2}\,\mathrm{Det}[\Sigma]^{1/2}}\,\mathrm{Exp}\left[-\frac{1}{2}\,(x-\mu)^T\,\Sigma^{-1}\,(x-\mu)\right] \qquad (2)$$

We define a 2-variate density:

In[25]:=

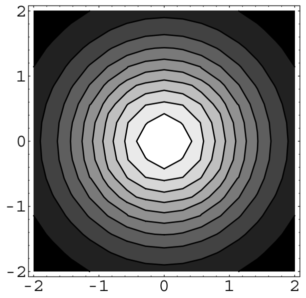
```
multigaus[x_,m_,cov_]:=
    Module[{IC,detCov,norm,p},
    IC = Inverse[cov];
    detCov = Abs[Det[cov]];
    norm = N[Sqrt[(2Pi)^2 detCov]];
    p = Exp[-0.5 (x-m).IC.(x-m)]/norm;
    Return[p];
];
```

Two variable examples

Zero mean, zero correlation

In[26]:=

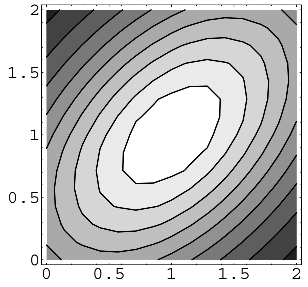
```
m1 = {0,0};
Cov = {{1,0},{0,1}};
ContourPlot[multigaus[{x1,x2},m1,Cov],{x1,-2,2}, {x2,-2,2}];
```



Mean = {1,1}, positive correlation

In[29]:=

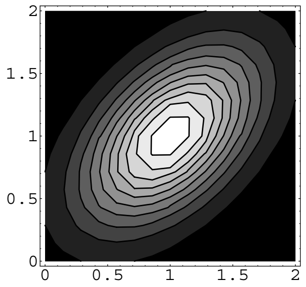
```
m1 = {1,1};
Cov = {{1,.5},{.5,1}};
ContourPlot[multigaus[{x1,x2},m1,Cov],{x1,0,2}, {x2,0,2}];
```



Mean = {1,1}, positive correlation, small variance

In[32]:=

```
m1 = {1,1};
Cov = 0.2{{1,.5},{.5,1}};
ContourPlot[multigaus[{x1,x2},m1,Cov],{x1,0,2}, {x2,0,2}];
```



Mixture of gaussians

α is a mixing parameter

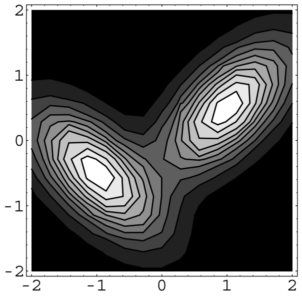
$$p(x) = \alpha p_1(x) + (1-\alpha)\, p_2(x) \text{ where } 0 \le \alpha \le 1 \qquad (3)$$

α can be interpreted in terms of a prior probability of choosing which of two distributions a sample will be drawn from.

In[35]:=

```
m1 = {1,.5}; m2 = {-1,-.5};
Cov1 = 0.4*{{1,.6},{.6,1}};
Cov2 = 0.4*{{1,-.6},{-.6,1}};
mix[x_] := 0.5 (multigaus[x,m1,Cov1] + multigaus[x,m2,Cov2]);
```

In[39]:=
```
ContourPlot[mix[{x1,x2}],{x1,-2,2}, {x2,-2,2}];
```



■ Drawing samples from the density--draw from a hat method

■ We'll simulate the process of filling a hat with slips of paper, where the number of slips is proportional to the probability the number being in some range (dx1,dx2)

```
m1 = {0,0};
Cov = {{1,.8},{.8,1}};
dx1 = 0.1;
dx2 = dx1;

Nslips=100;
hat = {};
For[x1=-2,x1<=2,x1=x1+dx1,
    For[x2=-2,x2<=2,x2=x2+dx2,
        np = Nslips*multigaus[{x1,x2},m1,Cov];
        For[i=1,i<np,i=i+1,
            hat = Append[hat,{x1,x2}];
        ];
    ];
];
```

hat is a list of pairs of numbers for which the frequency of occurence of pairs is determined by multigauss.

```
Dimensions[hat]
```

```
{8698, 2}
```

■ Now let's do a check, where we compile a histogram representing the frequencies of each slip

First, define the "bins" in domain, that we'll use to check for matches:

```
domain = {};
For[x1=-2,x1<=2,x1=x1+dx1,
    For[x2=-2,x2<=2,x2=x2+dx2,
            domain = Append[domain,{x1,x2}];
    ];
];
```

An alternate way of specifying the domain using Outer[], and Range[]:

```
domain2 = Flatten[Outer[List,Range[-2,2,dx1],Range[-2,2,dx1]],1];
```

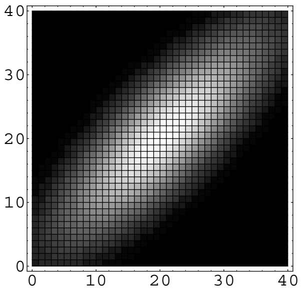
Now we'll count how many times we find that an element of hat matches a domain element:

```
Freq = Map[Count[hat,#]&,domain];
```

```
width = Sqrt[Dimensions[Freq]]
```

```
{40}
```

```
ListDensityPlot[Partition[Freq,width]];
```

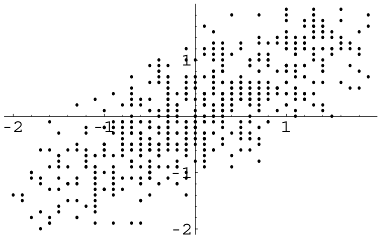


■ Now draw a sample--simulate pulling a slip from hat

```
rv:=hat[[Random[Integer,{1,Length[hat]}]]];
```

```
test = Table[hat[[Random[Integer,{1,Length[hat]}]]],{600}];
```

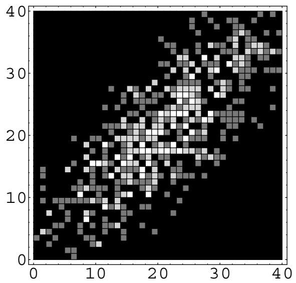
```
g1=ListPlot[test];
```



Of course, we can't see the frequency of draws in this plot, so let's count up the number of occurences per bin, and plot up the results as we did above.

```
Freq2 = Map[Count[test,#]&,domain];
width = Sqrt[Dimensions[Freq2]];
```

```
ListDensityPlot[Partition[Freq2,width]];
```



Drawing multivariate samples from the density -- use the inverse cumulative distribution

Add-on *Mathematica* functions for gaussian multivariates & exploring marginals

■ Define PDF, CDF

In[40]:=

```
m1 = {1,.5};
r=0.4*{{1,.6},{.6,4}};
ndist = MultinormalDistribution[m1, r];
```

In[43]:=

```
pdf = PDF[ndist, {x1, x2}]
```

Out[43]=

$$0.20855\, e^{\frac{1}{2}\,(-(-1+x1)\,(2.74725\,(-1+x1)-0.412088\,(-0.5+x2))-(-0.412088\,(-1+x1)+0.686813\,(-0.5+x2))\,(-0.5+x2))}$$

What is the probability of the distribution in the region $x_1 < -2 \bigcap x_2 < 1$.

In[44]:=

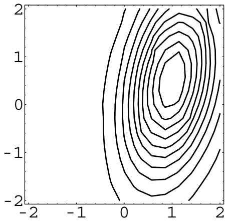
```
CDF[ndist, {-2, 1}]
```

Out[44]=

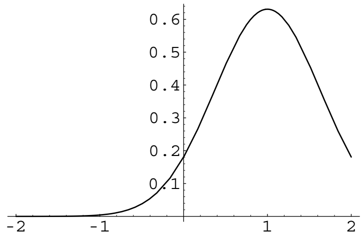
$$1.02471\times 10^{-6}$$

In[46]:=

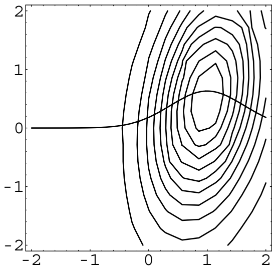
```
g1=ContourPlot[PDF[ndist, {x1, x2}],{x1,-2,2}, {x2,-2,2},ContourShading-
>False];
```



```
In[47]:= marginal[x1_] := Integrate[PDF[ndist, {x1, x2}], {x2, -Infinity, Infinity}] ;
         g2 = Plot[marginal[x1], {x1, -2, 2}];
```



```
In[49]:= Show[{g1, g2}];
```



■ Drawing samples

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

```
Random[ndist]
```

```
{-0.52553, -2.54926}
```

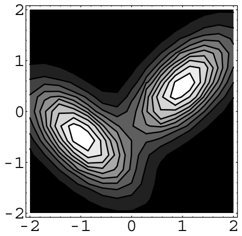
Mixtures of gaussians

```
Clear[mix];
```

```
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];
```

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

```
ContourPlot[mix[{x1,x2}],{x1,-2,2}, {x2,-2,2}];
```

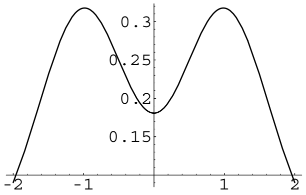


■ Marginals for mixture

```
marginal[x1_] := Integrate[mix[{x1, x2}], {x2, -Infinity, Infinity}]        (4)
```

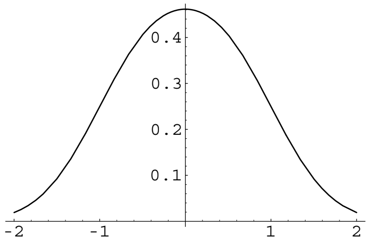
```
Clear[marginal];
marginal[x1_] :=
  0.5 * (Integrate[PDF[ndist1, {x1, x2}], {x2, -Infinity, Infinity}] +
     Integrate[PDF[ndist2, {x1, x2}], {x2, -Infinity, Infinity}]);
```

```
Plot[marginal[x1], {x1, -2, 2}];
```



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

```
Plot[marginal[x2], {x2, -2, 2}];
```



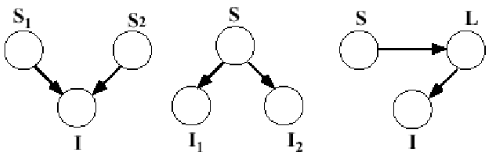
Which projection (marginal) is more "interesting"--the one onto x1 or onto x2?

Exploratory projection pursuit

Graphical Models of dependence

■ Graphs: causal structure and conditional independence

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 just three basic building blocks: converging, diverging, and intermediate nodes. For example, multiple causal variables causing a given measurement, a single variable producing multiple measurements, or a cause indirectly influencing a 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 data 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.

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

The data measurements (x) are determined by a typically non-linear function (ϕ) 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). In general, the causal structure of natural data (e.g. image or speech) patterns is more complex and consequently requires elaboration of its graphical representation. For 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): $\int_{\text{S\_g}} p(x, \text{S\_e}, \text{S\_g})\, d\text{S\_g}$.

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

Noise is whatever you don't care to estimate, but contributes to the data. The secondary variables are noise.

Optimal Inference and task dependence: Fruit example

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

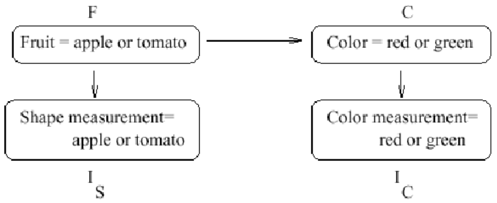

F
C
Fruit = apple or tomato
Color = red or green
Shape measurement= apple or tomato
Color measurement= red or green
I S
I C

Figure from Yuille, Coughlan, Kersten & Schrater.

The 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]

The prior model on hypotheses, F & C

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

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

```
F=a      9/16
F=t      7/16
```

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

```
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], {F, 1, 2}, {C, 1, 2}],
 TableHeadings -> {{"F=a", "F=t"}, {"C=r", "C=g"}}]
```

```
         C=r      C=g
F=a      5/9      4/9
F=t      6/7      1/7
```

So the joint is:

```
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"}}]
```

```
         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:

$$\texttt{ppC[C\_] :=} \sum_{F=1}^{2} \texttt{jpFC[F, C]}$$

Question: Is fruit identity independent of material color--i.e. is F independent of C?

■ Answer

No.

```
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"}}]
```

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

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

The generative model: Imaging probabilities

Analogous to collecting histograms for the two switch positions in the SDT experiment, 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]**:

```
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], {Ic, 1, 2}, {C, 1, 2}],
 TableHeadings -> {{"Ic=rm", "Ic=gm"}, {"C=r", "C=g"}}]
```

```
            C=r     C=g
Ic=rm       9/16    7/16
Ic=gm       1/2     1/2
```

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

```
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"}}]
```

```
            F=a     F=t
Is=am       11/16   5/8
Is=tm       5/16    3/8
```

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]:

```
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:

$$\texttt{jpIsIc[Is\_, Ic\_]} := \sum_{C=1}^{2} \sum_{F=1}^{2} \texttt{jpFCIsIc[F, C, Is, Ic]}$$

Three MAP tasks

Suppose that we measure Is=am, and Is = rm. The measurements suggest "red apple", but to find the most probable, we need to take into account the priors too.

■ Define argmax[] function:

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

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

Using the total joint, p(F,C | Is, Ic) = $\frac{p(F,C,\text{Is},\text{Ic})}{p(\text{Is},\text{Ic})} \propto$ p(F,C,Is,Ic)

```
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]
```

```
          C=r        C=g
F=a       495/4096   77/1024
F=t       135/1024   35/2048
```

```
135/1024
```

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

"Red tomato" is the most probable once we take into account the difference in priors.

Calculating p(F,C | Is, Ic). We didn't actually need p(F,C | Is, Ic), but we can calculate it by conditioning the total joint on the probability of the measurments:

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

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

```
         C=r       C=g
F=a     55/157    308/1413
F=t     60/157    70/1413
```

```
60/157
```

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

■ Pick most probable color--Answer "red"

In this case, we want maximize the posterior:

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

`pC[C_, Is_, Ic_] :=` $\sum_{F=1}^{2}$ `jpFCcIsIc[F, C, Is, Ic ]`

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

```
C=r     115/157
C=g     42/157
```

```
115/157
```

```
{{1}}
```

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

■ Pick most probable fruit--Answer "apple"

p(F | Is, Ic)

`pF[F_, Is_, Ic_] :=` $\sum_{C=1}^{2}$ `jpFCcIsIc[F, C, Is, Ic ]`

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

```
F=a     803/1413
F=t     610/1413
```

```
803/1413
```

```
{{1}}
```

The answer is "apple"

■ Moral of the story: Optimal inference depends on the precise definition of the task

Appendices

```
<< Graphics`Graphics`
```

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]
```

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

```
pxH pH
```

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

■ 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)
```

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

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

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

■ 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}}
```

$$\left\{\left\{\frac{1}{8}, \frac{1}{8}, \frac{1}{4}\right\}, \left\{\frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}\right\}$$

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.

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

```
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:

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

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

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

$$\begin{array}{llll} & x=1 & x=2 & x=3 \\ H=S1 & \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\ H=S2 & \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{array}$$

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]].**

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

```
1
```

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

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

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

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:

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

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[x|H]** is a **probability** function of x, but a **likelihood** function of H. The posterior probability is obtained by conditioning on x:

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

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:

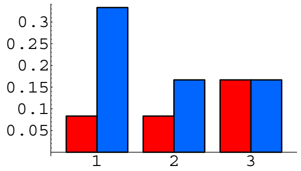
```
pHx = Transpose[Transpose[p] / px]
```

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

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];
```

```
TableForm[Table[p[H, x], {H, 1, 2}, {x, 1, 3}],
 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]].**

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

```
1
```

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

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

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

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

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

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

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

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

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

■ Conditioning

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

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

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

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

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

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

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[x|H]** is a **probability** function of x, but a **likelihood** function of H. The posterior probability is obtained by conditioning on x:

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

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

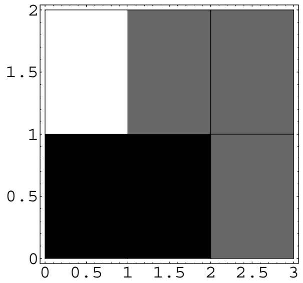
```
Table[pHx [H, x], {H, 1, 2}, {x, 1, 3}]
```

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

Plotting the joint

We use **ListDensityPlot[]**.

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



Exercises

Exercise: Use density mapping theorem to make random number generator for density p(y)

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.

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)

Knill, D. C., & Richards, W. (1996). *Perception as Bayesian Inference*. Cambridge: Cambridge University Press.

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

Van Trees, H. L. (1968). Detection, Estimation and Modulation Theory . New York: John Wiley and Sons.

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