Introduction to Neural Networks
U. Minn. Psy 5038

Gaussian generative models, learning, and inference

■ **Initialize standard library files:**

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

```
<< Statistics`MultinormalDistribution`
<< Statistics`DataManipulation`
```

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

Last time

Quick review of probability and statistics

Generative modeling: Drawing univariate samples

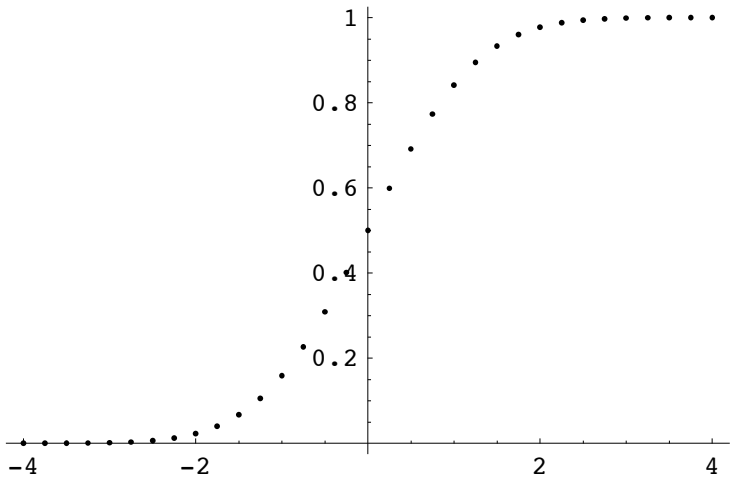
Example of look-up-table method that is fast and works for any continuous distribution.

■ **Cumulative distribution gaussian**

Suppose we have a list representing a cumulative distribution where the list is comprised of {x,y} pairs:

```
lcumulgauss = {{-4., 0.00003167124183305429},
   {-3.75, 0.00008841728520073352}, {-3.5, 0.00023262907903537412},
   {-3.25, 0.00057702504239064510}, {-3., 0.00134989803163002220},
   {-2.75, 0.00297976323505439900}, {-2.5, 0.00620966532577604100},
   {-2.25, 0.01224472655044533}, {-2., 0.02275013194817935800},
   {-1.75, 0.04005915686381704}, {-1.5, 0.06680720126885802},
   {-1.25, 0.10564977366685518}, {-1., 0.15865525393145696},
   {-0.75, 0.22662735237686812}, {-0.5, 0.30853753872598690},
   {-0.25, 0.40129367431707624}, {0., 0.5}, {0.25, 0.59870632568292370},
   {0.5, 0.69146246127401310}, {0.75, 0.77337264762313180},
   {1., 0.84134474606854300}, {1.25, 0.89435022633314480},
   {1.5, 0.93319279873114190}, {1.75, 0.95994084313618300},
   {2., 0.97724986805182060}, {2.25, 0.98777552734495540},
   {2.5, 0.99379033467422400}, {2.75, 0.99702023676494570},
   {3., 0.99865010196836990}, {3.25, 0.99942297495760930},
   {3.5, 0.99976737092096470}, {3.75, 0.99991158271479920},
   {4., 0.99996832875816690}};
```

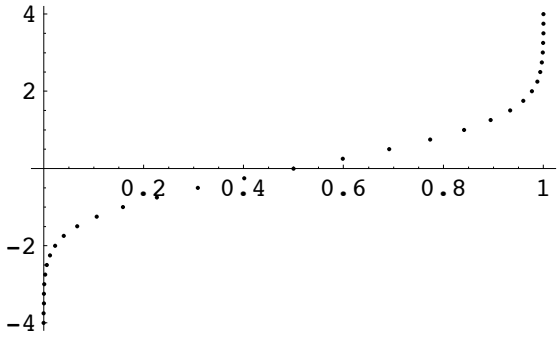
```
ListPlot[lcumulgauss];
```



■ Make inverse cumulative gaussian table

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

```
ListPlot[invlcumulgauss];
```

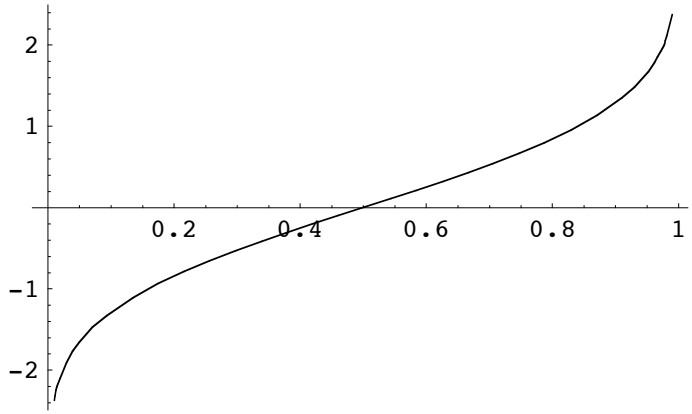


■ Make interpolated function of the inverse cumulative

If we pick Random[], this gives us a point on the x-axis, but it will almost certainly fall between the cracks. So we interpolate between the discrete points to get a continous function:

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

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



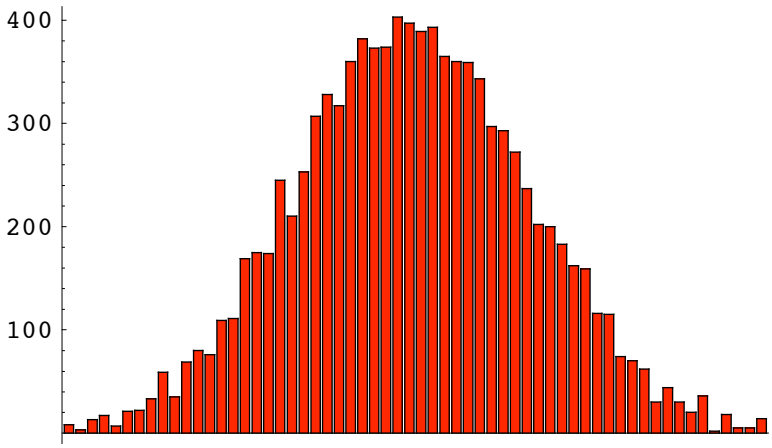
Interpolation works by fitting polynomial curves to the data. Try the test below with various interpolation orders (the default is 3)

```
test = Interpolation[{{1, 2.0}, {2, 4}, {3, 9}, {4, 16.0}},
   InterpolationOrder → 1];
Plot[test[x], {x, 1, 4}];
```

■ Draw a bunch of samples, and plot up histogram

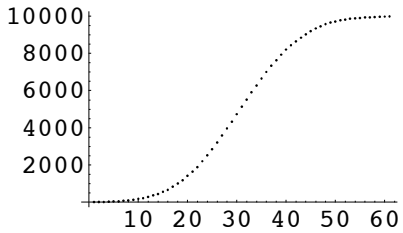
Now we are ready to draw 10,000 samples.

```
binsize = .1;
midpoints = Table[i + binsize / 2, {i, -3, 3 - binsize, binsize}];
zl = Table[interinvlcumulgauss[Random[]], {10000}];
freq = BinCounts[zl, {-3, 3, binsize}];
BarChart[Transpose[{freq, midpoints}], BarLabels → None];
```



■ Plot up cumulative histogram

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



Generative modeling: Multivariate gaussian, mixtures

■ 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:

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

We can use Add-on *Mathematica* functions for multivariate gaussians.

■ Define PDF, CDF

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

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

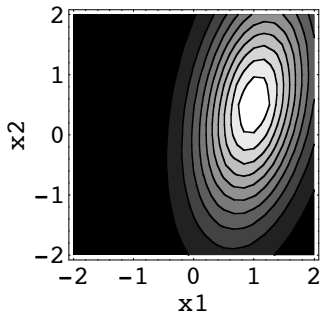
$$0.20855\, e^{\frac{1}{2}(-(x1-1)(2.74725(x1-1)-0.412088(x2-0.5))-(0.686813(x2-0.5)-0.412088(x1-1))(x2-0.5))}$$

What is the probability of $\{x_1, x_1\}$ lying in the region $x_1 < -2 \bigcap x_1 < 1$.

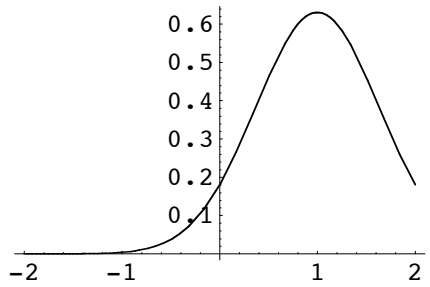
```
CDF[ndist, {-2, 1}]
```

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

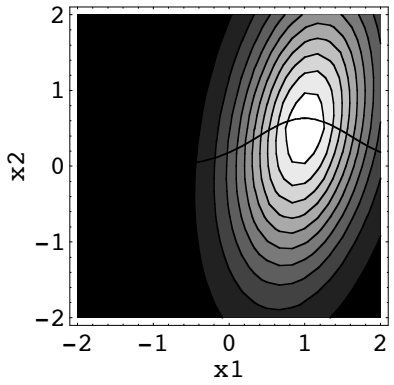
```
g1=ContourPlot[PDF[ndist, {x1, x2}],{x1,-2,2},
{x2,-2,2},FrameLabel->{"x1","x2"}];
```



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




```
Show[{g1, g2}];
```



■ Drawing samples

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

```
Random[ndist]
```

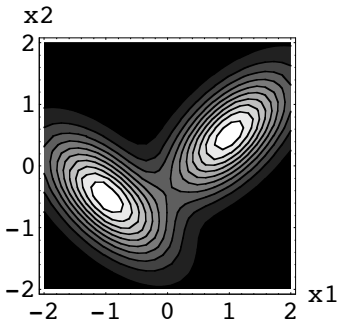
```
{2.30973, 0.674887}
```

■ Mixtures of gaussians

```
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}, PlotPoints->30,Axes→
True,AxesLabel->{"x1","x2"}];
```

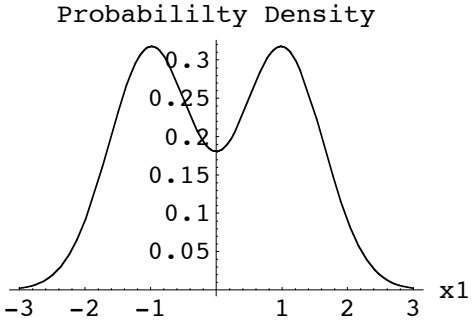


■ Marginals for mixture

```
marginal[x1_] := Integrate[mix[{x1, x2}], {x2, -Infinity, Infinity}]        (2)
```

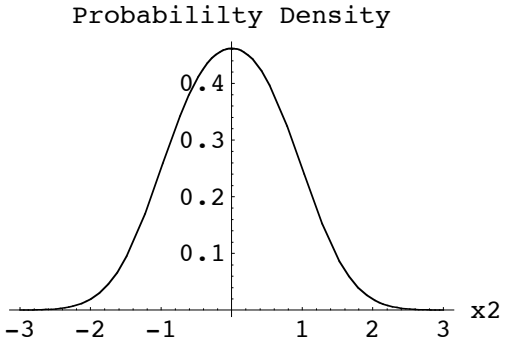
```
Clear[marginal];
marginal[x1_] :=
  0.5 * (NIntegrate[PDF[ndist1, {x1, x2}], {x2, -Infinity, Infinity}] +
     NIntegrate[PDF[ndist2, {x1, x2}], {x2, -Infinity, Infinity}]);
```

```
Plot[marginal[x1], {x1, -3, 3}, AxesLabel → {"x1", "Probabililty Density"}];
```




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

```
Plot[marginal[x2], {x2, -3, 3}, AxesLabel → {"x2", "Probabililty Density"}];
```



Side comments & where we'll see this again

■ Projection pursuit

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

Exploratory projection pursuit. (e.g. Intrator, 1993).

■ Inference: Learning parameters of mixture distributions

Return later to the inference problem: Given data, estimate the mixing parameters, means and covariances. EM algorithm.

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--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 (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.

$$x_i = \text{noise, where noise} \sim N[\mu, \sigma], \text{ or equivalently}$$
$$x_i = \mu + \text{noise, where noise} \sim N[0, \sigma] \quad (3)$$

```
ndist0 = NormalDistribution[μ, σ];
```

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

$$\mu \sim N[\mu 0, \sigma 0] \quad (4)$$

I.e. we make an initial guess of the mean's mean ($\mu 0$) and standard deviation ($\sigma 0$). But we are willing to change our estimate of the mean given new data--i.e. given the posterior. 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.

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

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

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:

$$p(\mu \mid x_1, x_2, x_3) = \frac{p(x_1, x_2, x_3 \mid \mu)\, p(\mu)}{p(x_1, x_2, x_3)} \quad (5)$$

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

```
PDF[ndist0, x1]
```

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

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 $p(\mu)$:

```
PDF[ndist0, x1] * PDF[ndist0, x2] * PDF[ndist0, x3] * PDF[ndistμ, μ]
```

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

■ 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:

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

$$-\frac{(\mu-\mu 0)^2}{2\sigma 0^2} - \text{Log}[4] - 2\,\text{Log}[\pi] - 3\,\text{Log}[\sigma] - \text{Log}[\sigma 0] - \frac{(-\mu+x_1)^2}{2\sigma^2} - \frac{(-\mu+x_2)^2}{2\sigma^2} - \frac{(-\mu+x_3)^2}{2\sigma^2}$$

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

$$\left\{\left\{\mu \to \frac{\frac{\mu 0}{\sigma 0^2}+\frac{x_1}{\sigma^2}+\frac{x_2}{\sigma^2}+\frac{x_3}{\sigma^2}}{\frac{3}{\sigma^2}+\frac{1}{\sigma 0^2}}\right\}\right\}$$

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

$$\left\{\left\{\mu \to \frac{\frac{\mu 0}{\sigma 0^2}+\frac{1}{\sigma^2}\sum_{i=1}^{n} x_i}{\frac{n}{\sigma^2}+\frac{1}{\sigma 0^2}}\right\}\right\} \tag{6}$$

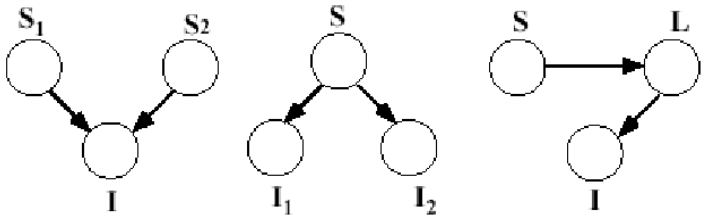
For the multi-variate case, see Duda and Hart.

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?

Graphical Models of dependence

Graphs: causal structure and conditional independence

The idea is to represent the probabilistic structure of a joint distribution, say of three random variables, P(S,L,I) by a Bayes net (e.g. Ripley, 1996}, which is a graphical model that expresses how variables influence each other. Let's consider 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,in visual perception, 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) .

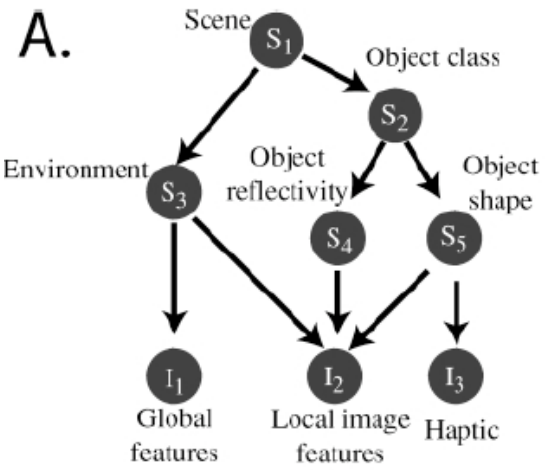
The arrows tell us how to factor the joint probability into conditionals. So for the three examples above, we have:

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)

Graphical models can provide a way of sketching out the conditional relationships between a complex set of interactions. Below is an example (see Kersten, Mamassian and Yuille, 2003) of graphical model for high-level vision.



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

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.

Two random variables can also be conditionally dependent which leads to a phenomenon sometimes referred to as "explaining away".

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

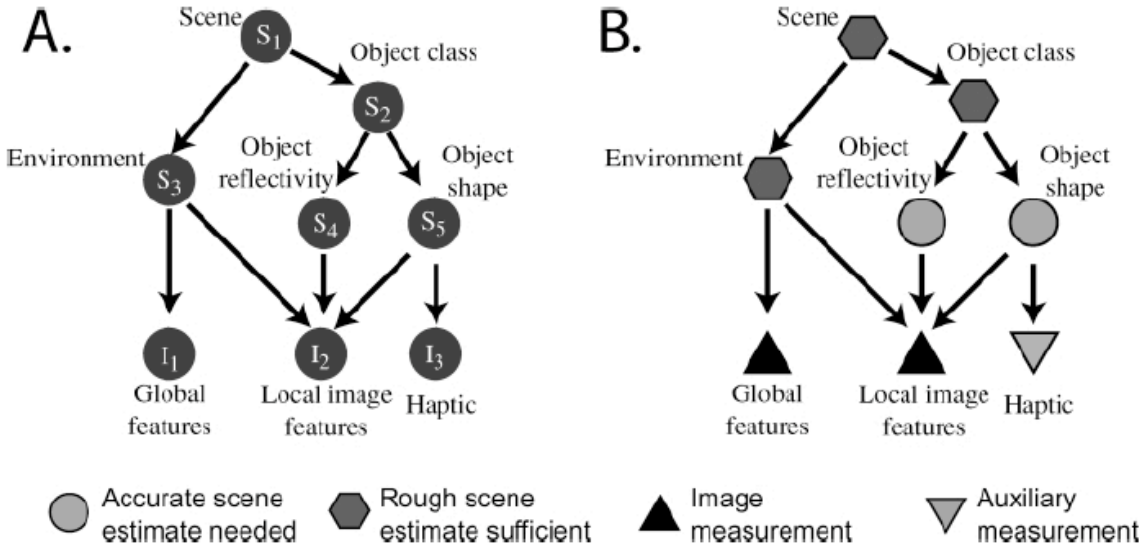
The task: Marginalization, primary and secondary variables

Data measurements are typically of function of variables we'd like to estimate and confounding variables. The task determines what we want to estimate, and we call these the primary or signal variables. The confounding variables we call secondary (or "nuisance" or "generic" or "noise" variables).

The data measurements (I) are determined by a typically non-linear function (ϕ) of primary signal variables (S_e) (to be explicitly estimated) and confounding secondary variables (S_g) (also called "generic" variables to be discounted).

Knowledge is represented by the joint probability p(I, 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 primary signal variables, while discounting the noise or secondary variables. Thus optimal perceptual decisions are determined by p(I,S_e), which is derived by summing over the secondary variables (i.e. marginalizing with respect to the secondary variables): $\int_{S\_g} p(I, S\_e, S\_g)\, d\,S\_g$.

The primary variables need to be estimated accurately. Noise is whatever you don't care to estimate, but contributes to the data. The secondary variables are noise, where either no estimate is required, or perhaps only a rough estimate.



Optimal inference: Putting things together

In theory, knowledge about a problem can be represented by a joint probability distribution involving three types of variables: the primary causes (S_e), the secondary causes (S_g), and the effect of the causes, the data (I). What is primary and what is secondary depends on the definition of the task. Variables that are primary for one task can be secondary for another.

Optimal inference is based on the distribution that you get by *conditioing on what you know:* p(I,S_e,S_g)/p(I) = p(S_e,-S_gI), and *marginalizing with respect to the variables you don't care about*: $\int_{S\_g} p(S\_e, S\_g \mid I) / p(I)\, d S\_g$.

The result is a posterior term that only includes the data and the primary variables: p(,S_eI). With this, we can ask questions like: given I, what value of S_e makes the posterior the biggest?

Optimal Inference and task dependence: Fruit example

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

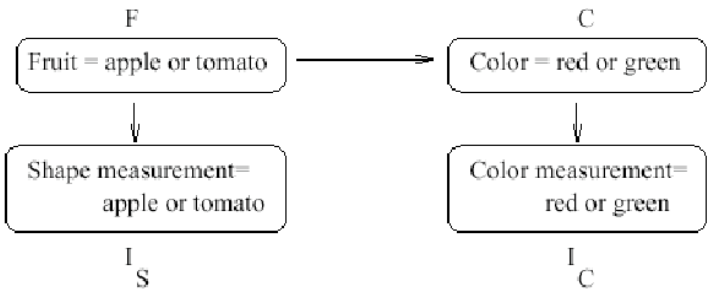


Figure from Yuille, Coughlan, Kersten & Schrater.

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]

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	$\frac{9}{16}$
F=t	$\frac{7}{16}$

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	$\frac{5}{9}$	$\frac{4}{9}$
F=t	$\frac{6}{7}$	$\frac{1}{7}$

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	$\frac{5}{16}$	$\frac{1}{4}$
F=t	$\frac{3}{8}$	$\frac{1}{16}$

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

```
ppC[C_] := Sum[jpFC[F, C], {F, 1, 2}]
```

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	$\frac{5}{16}$	$\frac{1}{4}$
F=t	$\frac{3}{8}$	$\frac{1}{16}$

	C=r	C=g
F=a	$\frac{99}{256}$	$\frac{45}{256}$
F=t	$\frac{77}{256}$	$\frac{35}{256}$

The generative model: Imaging probabilities

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	$\frac{9}{16}$	$\frac{7}{16}$
Ic=gm	$\frac{1}{2}$	$\frac{1}{2}$

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	$\frac{11}{16}$	$\frac{5}{8}$
Is=tm	$\frac{5}{16}$	$\frac{3}{8}$

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:

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

Three MAP tasks

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

■ Define argmax[] function:

```
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-like 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.

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=1, Ic=1)

```
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	$\frac{495}{4096}$	$\frac{77}{1024}$
F=t	$\frac{135}{1024}$	$\frac{35}{2048}$

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

```
{{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	$\frac{55}{157}$	$\frac{308}{1413}$
F=t	$\frac{60}{157}$	$\frac{70}{1413}$

$$\frac{60}{157}$$

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

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

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

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

C=r	$\frac{115}{157}$
C=g	$\frac{42}{157}$

$$\frac{115}{157}$$

```
{{1}}
```

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)

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

```
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". 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.

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 = { 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 = {{1/12, 1/12, 1/6}, {1/3, 1/6, 1/6}}
```

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

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

	x=1	x=2	x=3
H=S1	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{6}$
H=S2	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{6}$

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]]
```

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

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:

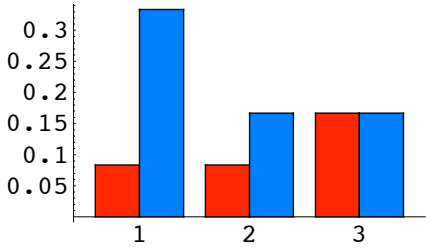
```
pHx = Transpose[Transpose[p] / px]
```

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

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	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{6}$
H=s2	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{6}$

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];
```

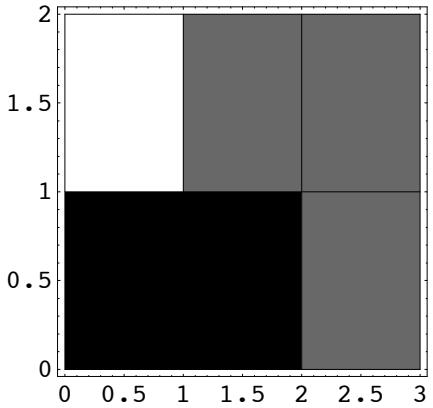
```
Table[pHx [H, x], {H, 1, 2}, {x, 1, 3}]
```

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

Plotting the joint

We use **ListDensityPlot[]**.

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



■ 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

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

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