Computational Vision
U. Minn. Psy 5036
Daniel Kersten
Lecture 6

■ Initialize standard library files:

```
<< Graphics`Graphics3D`
<< Statistics`MultinormalDistribution`
<< Graphics`ImplicitPlot`
```

```
<< Statistics`NormalDistribution`
Off[General::spell1];
```

Goals

Last time

Developed signal detection theory for characterizing an ideal observer for detecting a "known" pattern in additive gaussian noise.

The statistical treatment is a special case of Bayesian inference.

Showed how human and ideal performance can be quantitatively compared by their respective sensitivities, d's.

This time

Extend the tools of signal detection theory to "pattern inference theory", to begin to deal with more complex tasks.

How to manage complex pattern inference tasks?

■ Two main observations for simplification:

Graphical models of influence

Task dependence: Bayesian inference theory -> Bayesian decision theory, to take into account what information is important and what is not. I.e. what is signal and what is noise.

(Related to "integrating out", below.)

Graphical Models of dependence

The generative model in the previous lecture was simple. The signals were two fixed images (e.g. a sinusoidal grating and a uniform image), and the image variability was solely due to additive noise. What about natural images?

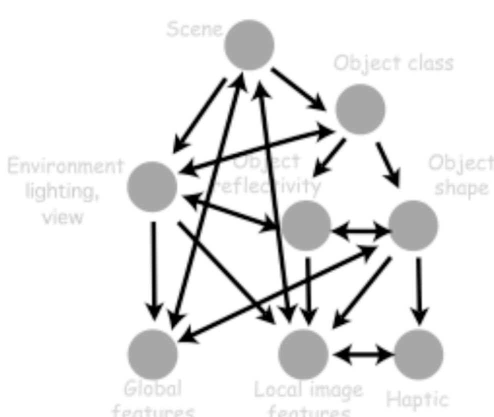
The "universe" of possible factors generating an image could be expressed by constructing the joint probability on all possible combinations of descriptions:

p(scene,object class, environment lighting,object reflectivity, object shape, global features, local features, haptic)

where each of the variable classes is itself a high-dimensional description, but this is hopelessly large, because of the combinatorial problem.

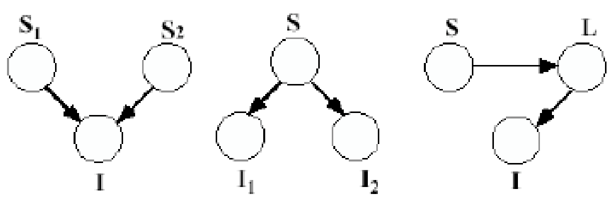
Natural images are complex, and in general it is difficult and often impractical to build a detailed quantitative generative model.But natural images do have regularities, and we can get insight into the problem by considering how various factors might produce natural images.

One way to begin simplifying the problem is to note that not all variables have a direct influence on each other. So draw a graph in which lines only connect variables that influence each other. In particular, we will use directed graphs to represent conditional probabilities.



Graphs: causal structure and conditional independence

The idea is that natural image pattern formation is specified by a high-dimensional joint probability, requiring an elaboration of the causal structure that is more complex than the simple SDT model. We 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 (random) 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) .

A basic rule of probability is the product rule, in which the joint probability p(A,B) = p(A|B)p(B) (see Probability-Overview.nb).

The arrows above is a graphical shorthand that tells us how to factor a 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)

Basic rules: Condition on what is known, and integrate out what you don't care about

■ Condition on what is known:

Given a scene description S, and image features I, the "universe" of possibilities is:

$$p(S, I) \tag{1}$$

If we know (i.e. the visual system has measured some image feature I), the joint can be turned into a conditional (posterior):

$$p(S \mid I) = p(S, I) / p(I) \tag{2}$$

■ Integrate out what you don't care about

We don't care to estimate the noise (or other generic, nuisance, or secondary variables):

$$p\,(S_{signal} \mid I) = \sum_{S_{noise}} p\,(S_{signal}, S_{noise} \mid I),$$
$$\text{or if continuous} = \int_{S_{noise}} p\,(S_{signal}, S_{noise} \mid I)\, dS_{noise} \tag{3}$$

Called "integrating out" or "marginalization"

Graphical models and general inference

■ Three types of nodes in a graphical model: known, unknown to be estimated, unknown to be integrated out (marginalized)

We have three basic states for nodes in a graphical model:

known

unknown to be estimated

unknown to be integrated out (marginalization).

We have causal state of the world S, that gets mapped to some image data I, perhaps through some intermediate parameters L, i.e. S->L->I.

So for example, face identity S determines facial shape L, which in turn determines the image I itself. Consider three very broad types of task:

■ Image data inference: synthesis

Image synthesis (forward, generative model): We want to model I through p(I|S). In our example, we want to specify "Bill", and then p(I|S="Bill") can be implemented as an algorithm to spit out images of Bill. If there is an intermediate variable, L, it gets integrated out.

■ Hypothesis ("inverse") inference

Hypothesis inference: we want to model samples for S: p(S|I).Given an image,we want to spit out likely object identities,so that we can minimize risk,or do MAP classification for accurate object identification. Again there is an intermediate variable, L, it gets integrated out. (Although we didnt' set up our SDT examples with dots and grating patterns to require explicit integrating out of the noise variables, it could be done that way.)

As emphasized before, the primary *focus of this course is to treat perception as "inverse" inference*.

■ Learning (parameter inference)

Although we have not described it in this lecture, **learning** can also be viewed as estimation**:** we want to model L: p(L|I,S), to learn how the intermediate variables are distributed. Given lots of samples of objects and their images, we want to learn the mapping parameters between them. (Alternatively, do a mental switch and consider a neural network in which an input S gets mapped to an output I through intermediate variables L. We can think of L as representing synaptic weights to be learned.)

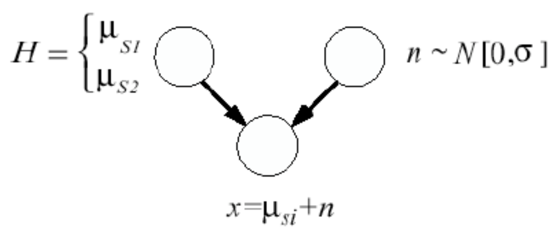
Two basic examples in standard statistics are:

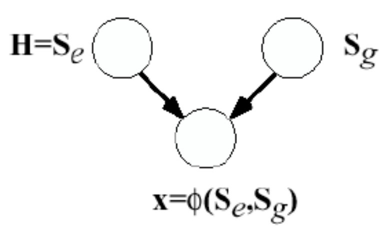
Regression: estimating parameters that provide a good fit to data. E.g. slope and intercept for a straight line through points $\{x_i, y_i\}$.

Density estimation: Regression on a probability density functions, with the added condition that the area under the fitted curve must sum to one.

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.





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 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). 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):

$$p(x, S_e) = \int_{S_g} p(x, S_e, S_g)\, dS_g \quad (4)$$

■ Conditional dependence and independence

Influences between variables are represented by conditioning, and a graphical model expresses the conditional independencies between variables. Two random variables may only become independent, once the value of some third variable is known. This is called conditional independence. From the probability overview, you note 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.

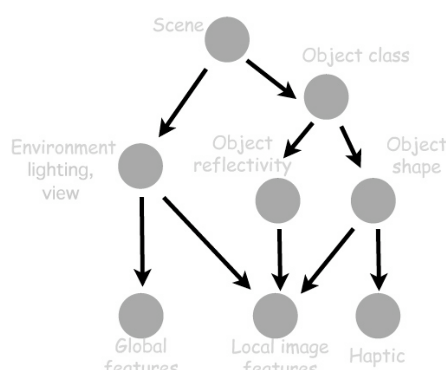
There is a correlation between eating ice cream and drowning. Why? What event should you condition on to make the dependence go away?

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

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

More complex generative and inference tasks

Generalize the notion of discounting



Some basic graph types in vision

From: Kersten, D., & Yuille, A. (2003). Bayesian models of object perception. *Current Opinion in Neurobiology, 13*(2), 1-9.

■ Basic Bayes

$$p[S \mid I] = \frac{p[I \mid S]\, p[S]}{p[I]}$$

Usually, we will be thinking of the **Y** term as a random variable over the hypothesis space, and **X** as data. So for visual inference, **Y = S** (the scene), and **X = I** (the image data), and **I = f(S)**.

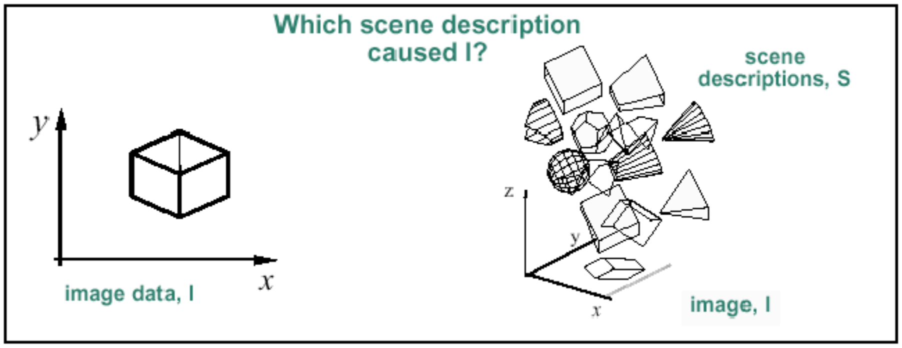
We'd like to have:

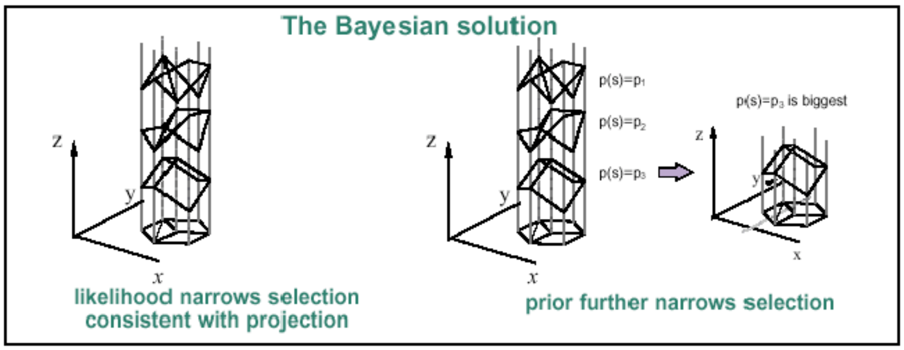
p(S|I), where is the **posterior** probability of the scene given the image

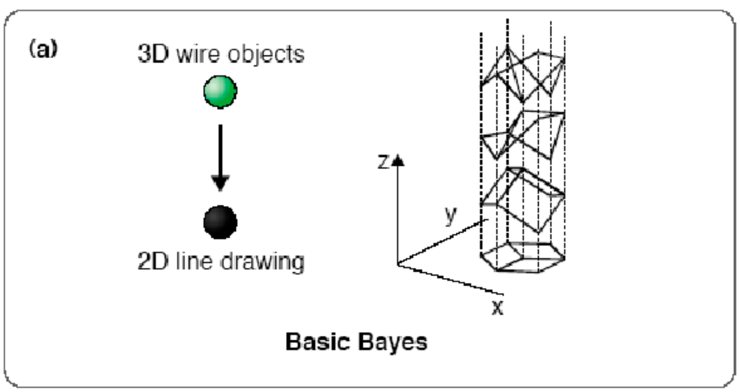
-- i.e. what you get when you condition the joint by the image data. The posterior is often what we'd like to base our decisions on, because as we discuss below, picking the hypothesis **S** which maximizes the posterior (i.e. maximum a posteriori or **MAP** estimation) minimizes the average probability of error.

p(S) is the **prior** probability of the scene.

p(I|S) is the **likelihood** of the scene. Note this is a probability of **I**, but not of **S**.

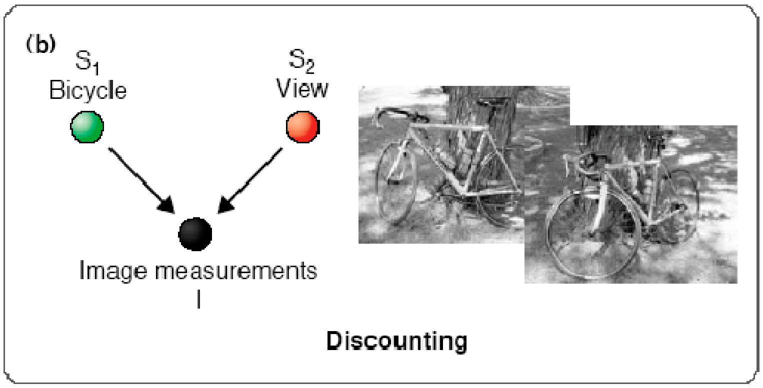






See: Sinha, P., & Adelson, E. (1993). Recovering reflectance and illumination in a world of painted polyhedra. Paper presented at the Proceedings of Fourth International Conference on Computer Vision, Berlin.

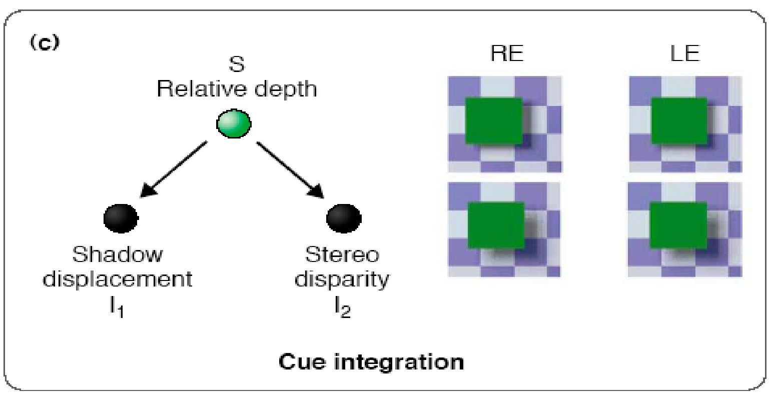
Discounting



The generative structure of the SDT problems we've looked at.

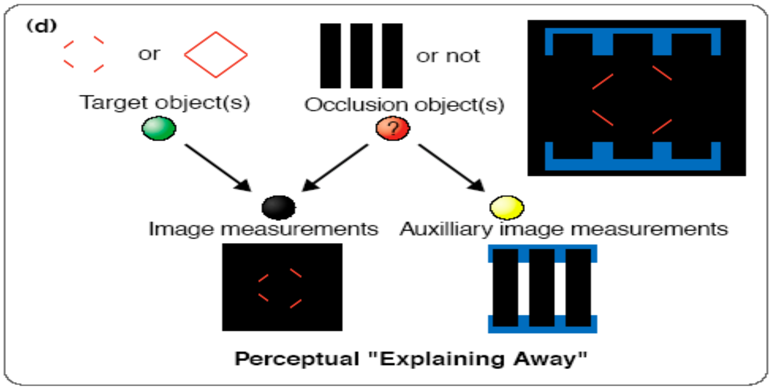
$$\mathrm{I}{=}\sum_{S_2} p(S_2, S_1 \mid I)$$

Cue integration



Here two measurements (shadow displacement and stereo disparity) may be correlated. However, if S is fixed, then they become *conditionally independent*.

■ Perceptual explaining away



The idea here is that one can have a probabilistic structure that gives rise to "competing explanations" for some image data.

This is a preview. We'll see more examples of this later in the course.

See: Lorenceau, J., & Shiffrar, M. (1992). The influence of terminators on motion integration across space. *Vision Res, 32*(2), 263-273.

Bayesian Decision Theory: Natural loss functions

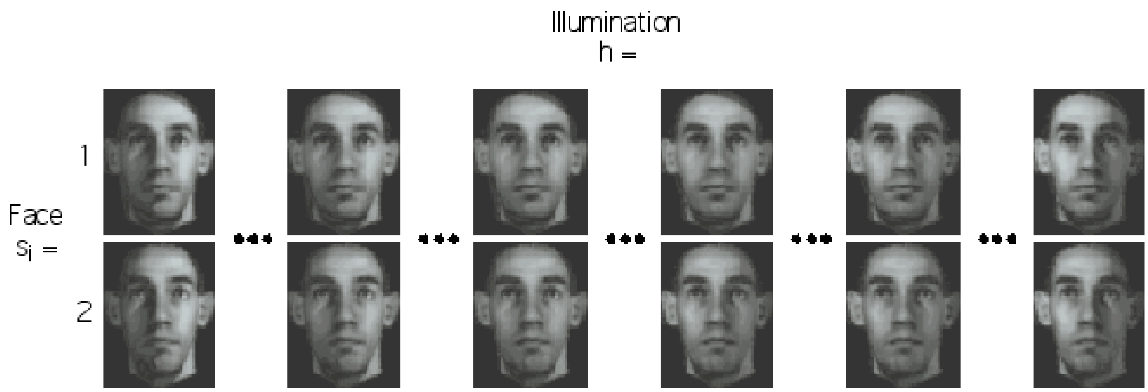
Bayes Decision theory, loss, and risk

We'd now like to generalize the idea of "integrating out" unwanted variables to allow us to put weights on how important a variable is for a task.

Earlier we noted that the costs of certain kinds of errors (e.g. a high cost to false alarms) could affect the decision criterion. Even though the sensitivity of the observer is essentially unchanged (e.g. the d' for the two Gaussian distributions remains unchanged), increasing the criterion can increase the overall error rate. This isn't necessarily bad.

A doctor might say that since stress EKG's have about a 30% false alarm rate, it isn't worth doing. The cost of a false alarm is high--at least for the HMO, with the resulting follow-ups, angiograms, etc.. And some increased risk to the patient of extra unnecessary tests. But, of course, false alarm rate isn't the whole story, and one should ask what the hit rate (or alternatively the miss rate) is? Miss rate is about 10%. (Thus, d' is actually pretty high--what is it?). From the patient's point of view, the cost of a miss is very high, one's life. So a patient's goal would not be to minimize errors (i.e. probability of a mis-diagnosis), but rather to minimize a measure of subjective cost that puts a very high cost on a miss, and low cost on a false alarm.

Although decision theory in vision has traditionally been applied to analogous trade-offs that are more cognitive than perceptual, recent work has shown that perception has implicit, unconscious trade-offs in the kinds of errors that are made.



One example is in shape from shading that we've seen before. An image provides the "test measurements" that can be used to estimate an object's shape and/or estimate the direction of illumination. Accurate object identification often depends crucially on an object's shape, and the illumination is a confounding (secondary) variable. This suggests that visual recognition should put a high cost to errors in shape perception, and lower costs on errors in illumination direction estimation. So the process of perceptual inference depends an task. The effect of marginalization in the fruit example illustrated task-dependence. Now we show how marginalization can be generalized through decision theory to model other kinds of goals than error minimization (MAP) in task-dependence.

Bayes Decision theory provides the means to model visual performance as a function of utility.

Some terminology. We've used the terms switch state, hypothesis, signal state as essentially the same--to represent the random variable indicating the state of the world--the "state space". So far, we've assumed that the decision, d, of the observer maps directly to state space, d->s. We now clearly distinguish the decision space from the state or hypothesis space, and introduce the idea of a loss L(d,s), which is the cost for making the decision d, when the actual state is s.

Often we can't directly measure s, and we can only infer it from observations. Thus, given an observation (image measurement) x, we define a risk function that represents the *average loss* over signal states s:

$$R(d; x) = \sum_s L(d, s)\, p(s \mid x) \tag{5}$$

This suggests a decision rule: $\alpha(x)=\underset{d}{\operatorname{argmin}}\, R(d;x)$. But not all x are equally likely. This decision rule minimizes the expected risk average over all observations:

$$R(\alpha) = \sum_x R(d; x)\, p(x) \tag{6}$$

We won't show them all here, but with suitable choices of likelihood, prior, and loss functions, we can derive standard estimation procedures (maximum likelihood, MAP, estimation of the mean) as special cases.

For the MAP estimator,

$$R(d; x) = \sum_s L(d, s)\, p(s \mid x) = \sum_s (1 - \delta_{d,s})\, p(s \mid x) = 1 - p(d \mid x) \tag{7}$$

where $\delta_{d,s}$ is the discrete analog to the Dirac delta function--it is zero if d≠s, and one if d=s.

Thus minimizing risk with the loss function L = $(1 - \delta_{d,s})$ is equivalent to maximizing the posterior, p(d|x).

What about marginalization? You can see from the definition of the risk function, that this corresponds to a uniform loss:

L = -1.

$$R(s1; x) = \sum_{s2} L(d2, s2)\, p(s1, s2 \mid x) \tag{8}$$

So for our face recognition example, a really huge error in illumination direction has the same cost as getting it right.For the fruit example, optimal classification of the fruit identity required marginalizing over fruit color--i.e. effectively treating fruit color identification errors as equally costly...even tho', doing MAP after marginalization effectively means we are not explicitly identifying color.

Graphical models for Hypothesis Inference: Three types

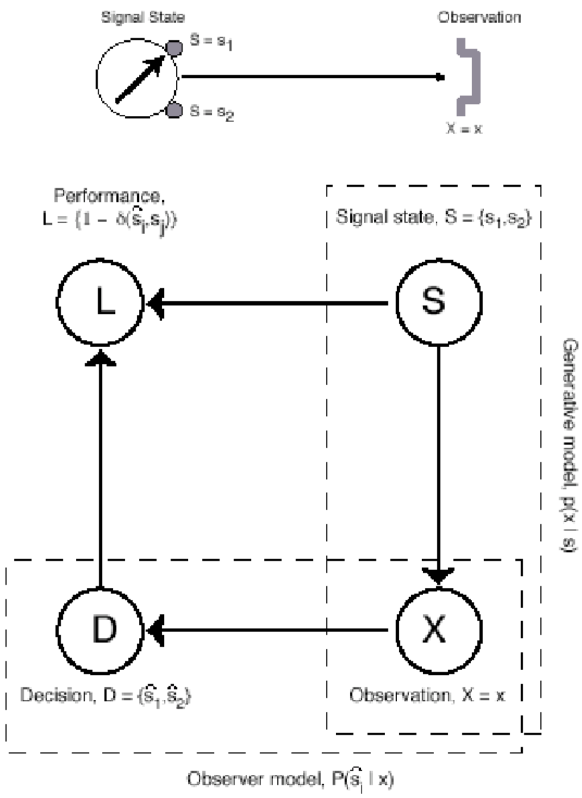
This section shows the common structure shared by three types of inference: detection, classification, and estimation.

Decisions can be right or wrong regarding a discrete hypothesis (detection, classification), or have some metric distance from an hypothesis along a continuous dimensions (estimation). Each decision or estimation has an associated loss function. There is a common graphical structure to each type of inference.

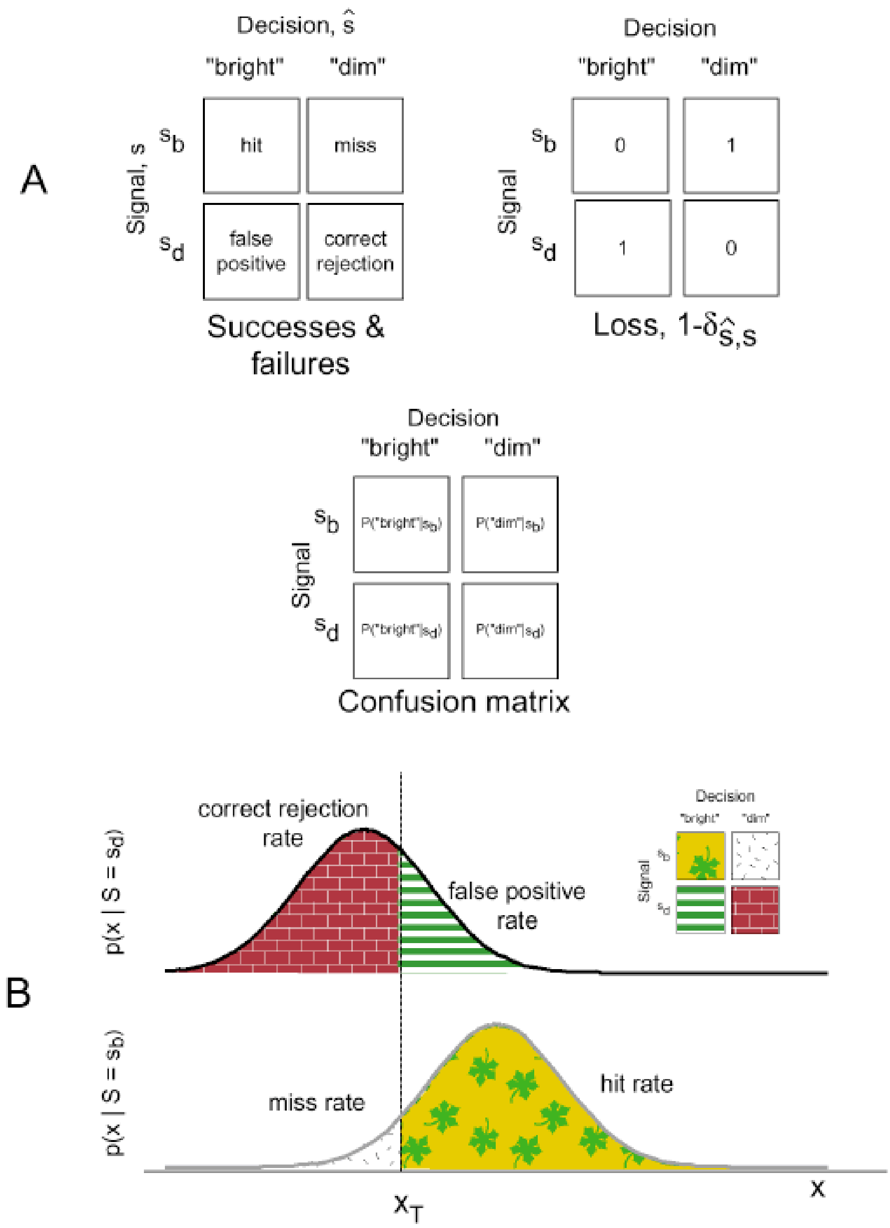
Hypothesis inference: Three types

■ Detection

Let the decision variable d, be represented by $\hat{s}$.

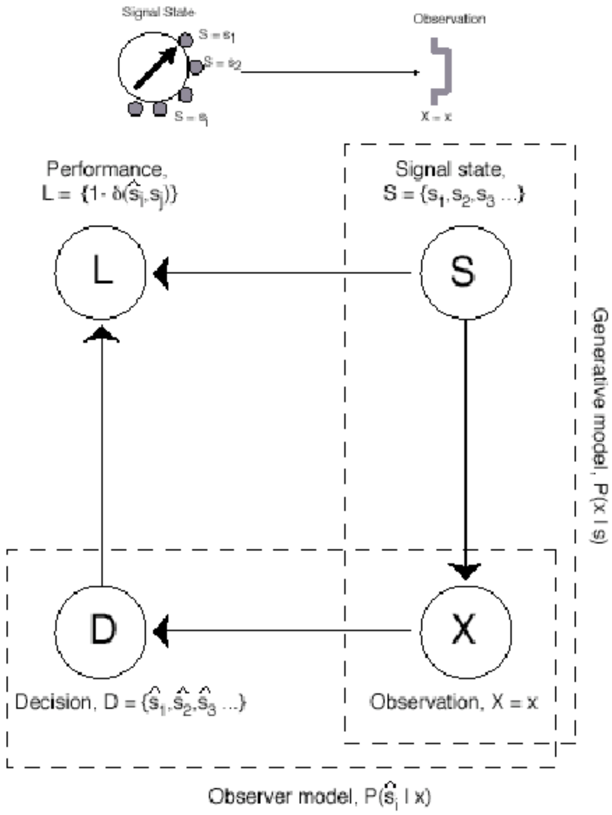

Signal State
S = s1
S = s2
Observation
X = x
Performance,
L = {1 − δ(ŝi,sj)}
Signal state, S = {s1,s2}
L
S
D
X
Generative model, p(x | s)
Decision, D = {ŝ1,ŝ2}
Observation, X = x
Observer model, P(ŝi | x)

■ loss function for yes/no task



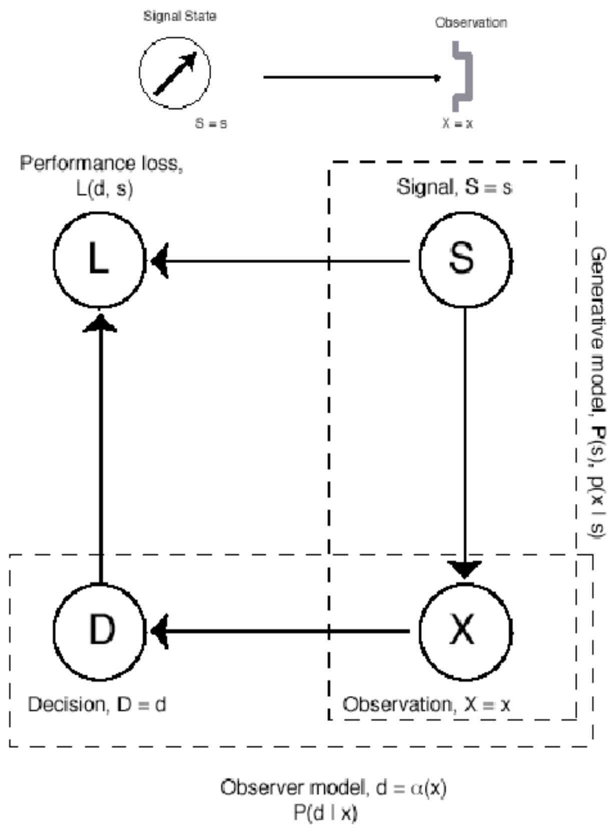
■ Classification

MAP rule: $\underset{i}{\operatorname{argmax}}\{p(S_i \mid x)\}$.



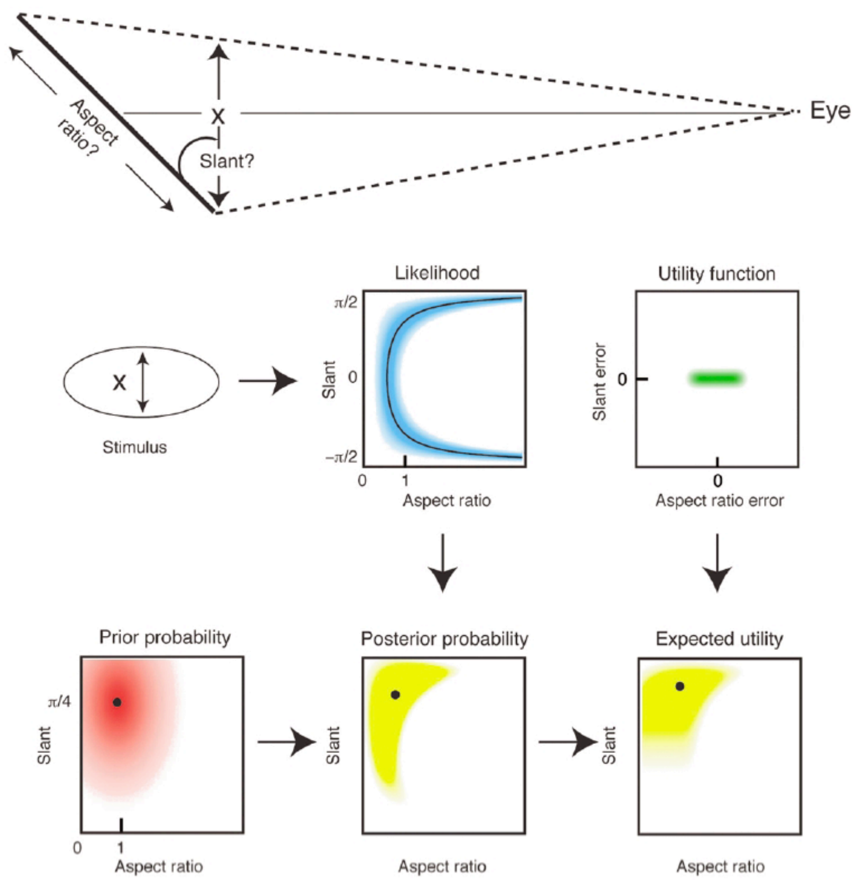
■ Continous estimation

$$\underset{S}{\text{argmax}}\{p(S \mid x)\}$$



One can show that L(d,s) = -(d-s)^2 produces an estimator that finds the mean, L(d,s) = -δ(d-s), does MAP (i.e. finds the mode), and L(d,s) = 1 is equivalent to marginalization (integrating out s).

Slant estimation example



From: Geisler, W. S., & Kersten, D. (2002). Illusions, perception and Bayes. *Nat Neurosci, 5*(6), 508-510.

Mathematica code to illustrate Bayesian estimation of surface slant and aspect ratio

This code was used to produce the figure in a Nature Neuroscience News & Views article by Geisler and Kersten (2002) that put in context a paper by Weiss, Simoncelli and Adelson.

Wilson S. Geisler and Daniel Kersten (2002) Illusions, perception and Bayes. Nature Neuroscience, 5 (6), 508-510. Or (pdf).

http://gandalf.psych.umn.edu/~kersten/kersten-lab/papers/GeislerKerstennn0602-508.pdf

http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12037517

For: Weiss, Y., Simoncelli, E. P., & Adelson, E. H. (2002). Motion illusions as optimal percepts. *Nat Neurosci, 5*(6), 598-604.

http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12021763

■ Initialization

```
<< Graphics`Graphics3D`
<< Statistics`MultinormalDistribution`
<< Graphics`ImplicitPlot`
```

```
npoints = 128;
loaspect = 0;
hiaspect = 5;
$TextStyle = {FontFamily → "Helvetica", FontSize → 14}
Fswitch = True;
```

```
{FontFamily → Helvetica, FontSize → 14}
```

```
PadMatrix[mat_, gray_, n_] := Module[{d},
   d = Dimensions[mat];
   Return[PadRight[PadLeft[mat, {d[[1]] + n, d[[2]] + n}, gray],
     {d[[1]] + 2 * n, d[[2]] + 2 * n}, gray]];
  ];
```

■ Init delta

```
gdelta[x_, w_] := 1 - (UnitStep[x + w / 2] - UnitStep[x - w / 2]);
(*Plot[gdelta[x,1],{x,-10,10},PlotRange→{0,2}];*)
```

■ Introduction

Consider the above figure.

Bayesian ideal observers for tasks involving the perception of objects or events that differ along two physical dimensions, such as aspect ratio and slant, size and distance, or speed and direction of motion. When a stimulus is received, the ideal observer computes the likelihood of receiving that stimulus for each possible pair of dimension values (that is, for each possible interpretation). It then multiplies this likelihood distribution by the prior probability distribution for each pair of values to obtain the posterior probability distribution—the probability of each possible pair of values given the stimulus. Finally, the posterior probability distribution is convolved with a utility function, representing the costs and benefits of different levels of perceptual accuracy, to obtain the expected utility associated with each possible interpretation. The ideal observer picks the interpretation that maximizes the expected utility. (Black dots and curves indicate the maxima in each of the plots.) As a tutorial example, the figure was constructed with a specific task in mind; namely, determining the aspect ratio and slant of a tilted ellipse from a measurement of the aspect ratio (x) of the image on the retina. The black curve in the likelihood plot shows the ridge of maximum likelihood corresponding to the combinations of slant and aspect ratio that are exactly consistent with x; the other non-zero likelihoods occur because of noise in the image and in the measurement of x. The prior probability distribution corresponds to the assumption that surface patches tend to be slanted away at the top and have aspect ratios closer to 1.0. The asymmetric utility function corresponds to the assumption that it is more important to have an accurate estimate of slant than aspect ratio.

■ Calculate Likelihood function and its maxima

$$p(I \mid S_{prim}, S_{\sec})$$

$$p(x \mid \alpha, d) = p(x - \phi(\alpha, d))$$

$$x = \phi(\alpha, d) + noise$$

Image model determines the constraint, x = d Cos[alpha] + noise, determines the likelihood

Assume noise has a Gaussian distribution with standard deviation = 1/5;

Assume an image measurement (x=1/2)

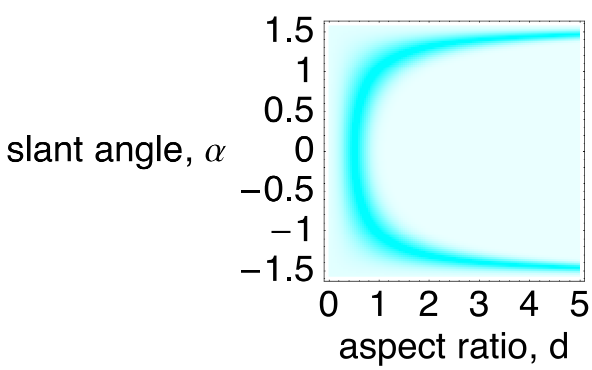
```
likeli[alpha_, x_, d_, s_] :=
 Exp[-((x - d Cos[alpha])^2) / (2 s^2)] (1 / Sqrt[2 Pi s^2])
likeli[α, x, d, s]
x = 1 / 2; s = 1 / 5;
like = likeli[α, x, d, s]
```

$$\frac{e^{-\frac{(0.2-d\cos(\alpha))^2}{2s^2}}}{\sqrt{2\pi}\sqrt{s^2}}$$

$$\frac{5\,e^{-\frac{25}{2}\left(\frac{1}{2}-d\cos(\alpha)\right)^2}}{\sqrt{2\pi}}$$

Plot likelihood

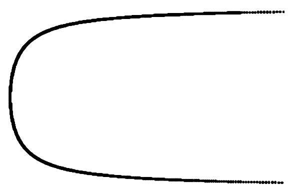
```
gdlike = DensityPlot[like, {d, loaspect, hiaspect}, {α, -Pi / 2, Pi / 2},
   PlotPoints → npoints, Mesh → False,
   ColorFunction -> (RGBColor[1 - (0.1 + 0.8 #), 1, 1] &),
   FrameLabel → {"aspect ratio, d", "slant angle, α"}, RotateLabel → False];
```



Plot likelihood maxima

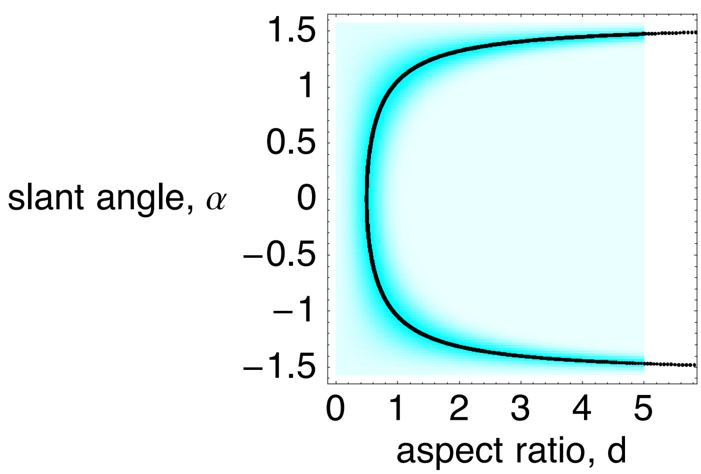
■ **There is no unique maximum. The likelihood function has a ridge**

```
temp2 = Table[Point[{x / Cos[alpha], alpha}], {alpha, -Pi / 2, Pi / 2, .001}];
temp =
 Join[Table[Point[{d, ArcCos[x / (d)]}], {d, loaspect + .5, hiaspect, .01}],
  temp2];
gtemp = Show[Graphics[{PointSize[.01], temp}]];
```



Plot likelihood together with maximum along the ridge

```
gdlike = DensityPlot[like, {d, loaspect, hiaspect}, {α, -Pi / 2, Pi / 2},
   PlotPoints → npoints, Mesh → False,
   ColorFunction -> (RGBColor[1 - (0.1 + 0.8 #), 1, 1] &),
   FrameLabel → {"aspect ratio, d", "slant angle, α"}, RotateLabel → False,
   Frame → Fswitch, DisplayFunction → Identity];
glikemax = Show[gdlike, gtemp, DisplayFunction → $DisplayFunction];
```



■ **Calculate the prior, and find its maximum**

$$p(S_{prim}, S_{sec})$$

$$p(\alpha, d)$$

The prior probability distribution corresponds to the assumption that surface patches tend to be slanted away at the top and have aspect ratios closer to 1.0. We model the prior by a bivariate gaussian:

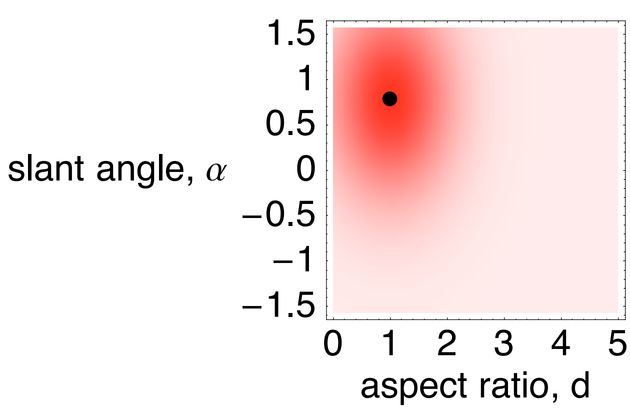
```
PDF[MultinormalDistribution[{μα, μd}, R], {α, d}]
```

$$\frac{e^{-\frac{1}{2}\{\alpha-\mu_\alpha, d-\mu_d\}.R^{-1}.\{\alpha-\mu_\alpha, d-\mu_d\}}}{2\pi\sqrt{|R|}}$$

```
R1 = {{.25, 0}, {0, .25}};
ndist3 = MultinormalDistribution[{Pi / 4., 1}, R1];
pdf3 = PDF[ndist3, {α, d}];
FindMinimum[-pdf3, {d, 5}, {α, 1}]
gdprior = DensityPlot[pdf3 ^ .4, {d, loaspect, hiaspect}, {α, -Pi / 2, Pi / 2},
   PlotPoints → npoints, Mesh → False,
   ColorFunction -> (RGBColor[1, 1 - (0.1 + 0.8 #), 1 - (0.1 + 0.8 #)] &),
   FrameLabel → {"aspect ratio, d", "slant angle, α"}, RotateLabel → False,
   DisplayFunction → Identity];
```

$\{-7.35283\times10^{-15}, \{d\to5., \alpha\to1.\}\}$

```
Show[gdprior, Graphics[{PointSize[.05], Point[{1, .785}]}],
  DisplayFunction → $DisplayFunction];
```



■ **Calculate the posterior, and find its maximum**

$$p(S_{prim}, S_{sec} \mid I) \propto p(I \mid S_{prim}, S_{sec})p(S_{prim}, S_{sec})$$

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

$$p(\alpha, d \mid x) \propto p(x \mid \alpha, d)p(\alpha, d)$$

More precisely, we'll calculate a quantity proportional to the posterior. The posterior is equal to the product of the likelihood and the prior, divided by the probability of the image measurement, x. Because the image measurement is fixed, we only need to calculate the product of the likelihood and the prior:

```
Clear[α, x, d, s];
likeli[α, x, d, s] * PDF[MultinormalDistribution[{μα, μd}, R], {α, d}]
```

$$\frac{e^{-\frac{(x-d\cos(\alpha))^2}{2s^2}-\frac{1}{2}\{\alpha-\mu_\alpha,d-\mu_d\}.R^{-1}.\{\alpha-\mu_\alpha,d-\mu_d\}}}{2\sqrt{2}\,\pi^{3/2}\sqrt{s^2}\,\sqrt{|R|}}$$

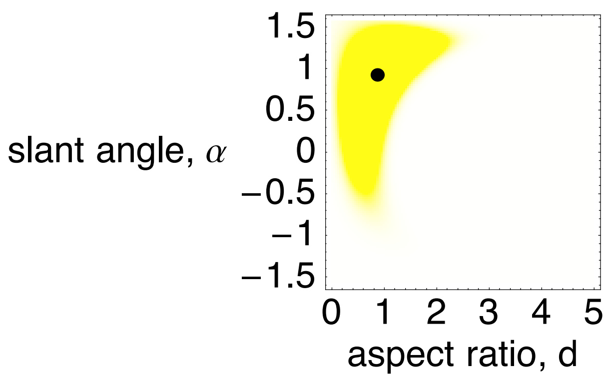
```
gdpost = DensityPlot[(pdf3 * like), {d, loaspect, hiaspect},
   {α, -Pi / 2, Pi / 2}, ColorFunction -> (RGBColor[1, 1, 1 - (0.01 + 0.9 #)] &),
   PlotPoints → npoints, Mesh → False,
   FrameLabel → {"aspect ratio, d", "slant angle, α"}, RotateLabel → False,
   Frame → Fswitch, DisplayFunction → Identity];
FindMinimum[-pdf3 * like, {d, {1, 3}}, {α, {.2, 1}}]
```

FindMinimum::nnum :

The function value $\{-0.0358912, -5.86992\times 10^{-11}\}$ is not a number at $\{d, \alpha\} = \begin{pmatrix} 1. & 3. \\ 0.2 & 1. \end{pmatrix}$. More…

FindMinimum[−pdf3 like, {d, {1, 3}}, {α, {0.2, 1}}]

```
Show[gdpost, Graphics[{PointSize[.05], Point[{.88, .92}]}],
  DisplayFunction → $DisplayFunction];
```



■ **Compute expected loss--i.e. risk, and find its minimum**

The expected loss is given by the convolution of the loss with the posterior:

risk=posterior*loss, where * means convolve; utility=-risk.

Loss function

$$l(\Delta\alpha, \Delta d) = l(\alpha' - \alpha, d' - d)$$

The asymmetric utility function corresponds to the assumption that it is more important to have an accurate estimate of slant than aspect ratio. The loss function reflects the task. Accurate estimates of slant may be more important for an action such as stepping or grasping, whereas an accurate estimation of aspect ratio may be more important for determining object shape (circular coffee mug top or not?).

```
maploss = Table[(1 - gdelta[x1d, .25]) * (1 - (gdelta[x2d, 2])),
   {x1d, -3, 3, 6 / npoints}, {x2d, -3, 3, 6 / npoints}];
gdloss = ListDensityPlot[maploss, Mesh → False,
   ColorFunction -> (RGBColor[1 - (0.01 + 0.9 #), 1 - (0.01 + 0.9 #), 1] &),
   Frame → False];
```


Convolve posterior with loss function

$$utility(\alpha', d') = -\sum_{\alpha, d} p(x \mid \alpha, d) p(\alpha, d) l(\alpha' - \alpha, d' - d)$$

Convert function description to numerical arrays for convolving

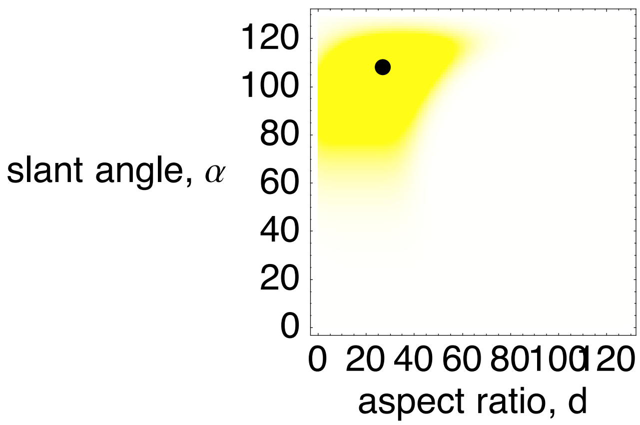
```
post =
  Transpose[Table[like * pdf3,
    {d, loaspect, hiaspect, (hiaspect - loaspect) / npoints},
    {α, -Pi / 2, Pi / 2, Pi / npoints}]];
post2 = PadMatrix[post, 0, 16];
maploss2 = PadMatrix[maploss, 0, 16];
offset = Floor[Dimensions[maploss2][[1]] / 2];
tempcon = ListConvolve[maploss2, post2, {-1, -1}];
risk2 = RotateLeft[tempcon, {offset, offset}];
risk = Take[risk2, {17, Dimensions[risk2][[1]] - 16},
   {17, Dimensions[risk2][[1]] - 16}];
```

```
grbrisk = ListDensityPlot[Map[#^1. &, risk], Mesh → False,
   ColorFunction -> (RGBColor[1, 1, 1 - (0.01 + 0.9 #)] &),
   FrameLabel → {"aspect ratio, d", "slant angle, α"}, RotateLabel → False,
   Frame → Fswitch, DisplayFunction → Identity];
```

```
Position[(risk), Max[(risk)]]
```

$\begin{pmatrix} 108 & 27 \end{pmatrix}$

```
Show[grbrisk, Graphics[{PointSize[.05], Point[{27, 108}]}],
  DisplayFunction → $DisplayFunction];
```



■ **Exercise: Compute and plot expected loss for the transposed loss function**

Inference: Fruit classification example

This section provides another quantitative example of inference. It illustrates how the task (i.e. what you integrate out) can change what is the optimal decision.

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

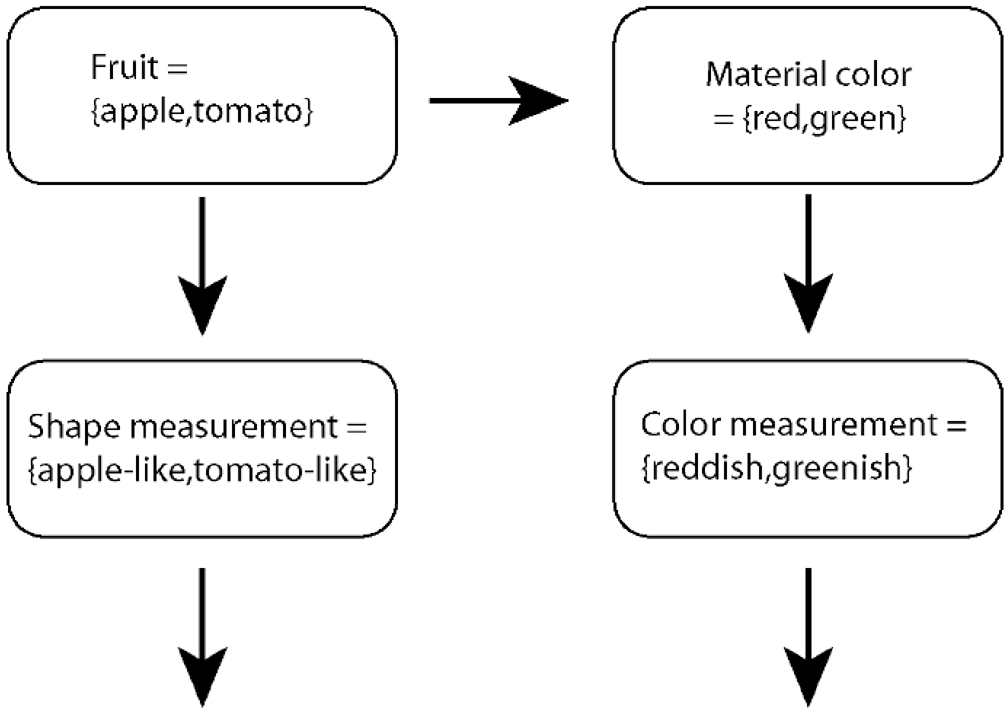


Figure from Yuille, Coughlan, Kersten & Schrater.

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	$\frac{9}{16}$
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:

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

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	$\frac{5}{16}$	$\frac{1}{4}$
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

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	$\frac{9}{16}$	$\frac{7}{16}$
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 ], {F, 1, 2}], {C, 1, 2}]
```

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

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

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

$(2 \quad 1)$

"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	$\frac{60}{157}$	$\frac{70}{1413}$

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

$(2 \quad 1)$

■ 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[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"

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	$\frac{803}{1413}$
F=t	$\frac{610}{1413}$

$$\frac{803}{1413}$$

(1)

The answer is "apple"

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

Exercises

MAP minimizes probability of error: Proof for detection

Here is why MAP minimizes average error. Suppose that x is fixed at a value for which P (S =sb |x)>P (S =sd |x). This is exactly like the problem of guessing “heads ”or “tails ” for a biased coin, say with a probability of heads P (S =sb|x). Imagine the light discrimination experiment repeated many times and you have to decide whether the switch was set to bright or not –but only on those trials for which you measured exactly x .The optimal strategy is to always say “bright ”. Let's see why. First note that:

p(error|x) = p(say "bright", actually dim |x) + p(say "dim", actually bright |x) =

p($\hat{s1}$, s2 | x)+p(s1, $\hat{s2}$ | x)

Given x, the response is independent of the actual signal state (see graphical model for detection above--"response is conditionally independent of signal state, given observation x"), so the joint probabilities factor:

p(error|x) = p(say "bright" |x)p(actually dim |x) + p(say "dim" |x)p(actually bright |x)

Let **t= p(say "bright" |x)**, then

p(error|t,x) = t*p(actually dim |x) + (1-t)*p(actually bright |x).

p(error|t), as a function of t, defines a straight line with slope **p(actually dim |x)-p(actually bright |x)**. (Just take the partial derivative with respect to t.) We've assumed P (S =sb |x)>P (S =s d |x)), so p(error|t) has a negative slope, with the smallest non-negative value of t being one. So, error is minimized when t=p(say "bright" |x)=1. I.e. Always say "bright".

Always saying "bright" results in a probability of error P (error |x)=P (S =sd |x).That's the best that can be done on average. On the other hand, if the observation is in a region for which P (S =sd|x)>P (S =sb |x),the minimum error strategy is to always pick "dim" with a resulting P (error |x)=P (S =sb |x).Of course, x isn't fixed from trial to trial, so we calculate the total probability of error which is determined by the specific values where signal states and decisions don 't agree:

$$\text{p(error)}=\sum_{i\neq j} p(\hat{s}_i, s_j)$$
$$=\sum_{i\neq j}\int p(\hat{s}_i, s_j \mid x)\, p(x)\, dx=\sum_{i\neq j}\int p(\hat{s}_i \mid x)\, p(s_j \mid x)\, p(x)\, dx$$

Because the MAP rule ensures that $p(\hat{s}_i, s_j \mid x)$ is the minimum for each x, the integral over all x minimizes the total probability of error.

Exercise: Show that MAP minimizes the probability of error for classification

$$\text{p(error)}=\sum_{i\neq j} p(\hat{s}_i, s_j)$$
$$=\sum_{i\neq j}\int p(\hat{s}_i, s_j \mid x)\, p(x)\, dx=\sum_{i\neq j}\int p(\hat{s}_i \mid x)\, p(s_j \mid x)\, p(x)\, dx$$

Let $\overset{\wedge *}{s}$ be the MAP choice, and the error using the MAP choice is,

$$\text{p(error*)}=\sum_{i\neq j} p\left(\overset{\wedge *}{s}_i, s_j\right)$$
$$=\sum_{i\neq j}\int p\left(\overset{\wedge *}{s}_i, s_j \mid x\right) p(x)\, dx=\sum_{i\neq j}\int p\left(\overset{\wedge *}{s}_i \,\middle|\, x\right) p(s_j \mid x)\, p(x)\, dx$$

We want to show that: $\text{p(error)-p(error*)}\geq 0$

Exercise: Show that MAP minimizes the probability of error for estimation

Show that R(d;x) is the average error rate for observation x,over all s.Then show that the risk R(α) is the expected number of errors over all x,when using the decision rule α(x).

Appendices

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.

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

$x \in \{1, 2, ...\}$

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	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{6}$
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:

$$p[x \mid H] = \frac{p[H,\ x]}{p[H]} \tag{9}$$

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:

$$p[\,H\mid x] = \frac{p[H,\ x]}{p[x]} \tag{10}$$

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]]];
```

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)]
```

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

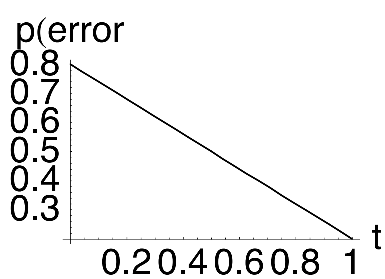
■ **Show that this joint probability has a uniform prior (i.e. both priors equal).**

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

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

Figure code

```
x = 0.2; y = 0.8;
Plot[t x + (1 - t) y, {t, 0, 1}, AxesLabel → {t, "p(error"}];
```



References

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

Green, D. M., & Swets, J. A. (1974). Signal Detection Theory and Psychophysics . Huntington, New York: Robert E. Krieger Publishing Company.

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.

Kersten, D., & Yuille, A. (2003). Bayesian models of object perception. *Current Opinion in Neurobiology, 13*(2), 1-9.

Kersten, D. (1999). High-level vision as statistical inference. In M. S. Gazzaniga (Ed.), *The New Cognitive Neurosciences -- 2nd Edition* (pp. 353-363). Cambridge, MA: MIT Press.

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.

Yuille, A. L., & Bülthoff, H. H. (1996). Bayesian decision theory and psychophysics. In K. D.C. & R. W. (Eds.), *Perception as Bayesian Inference*. Cambridge, U.K.: Cambridge University Press.

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