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

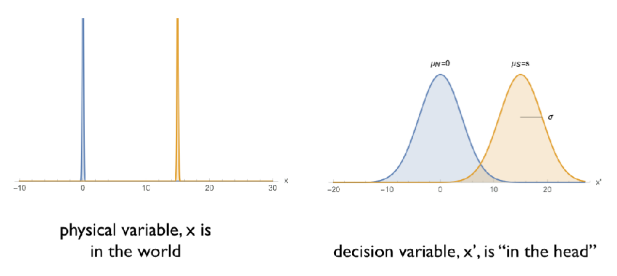
Goals

Last time

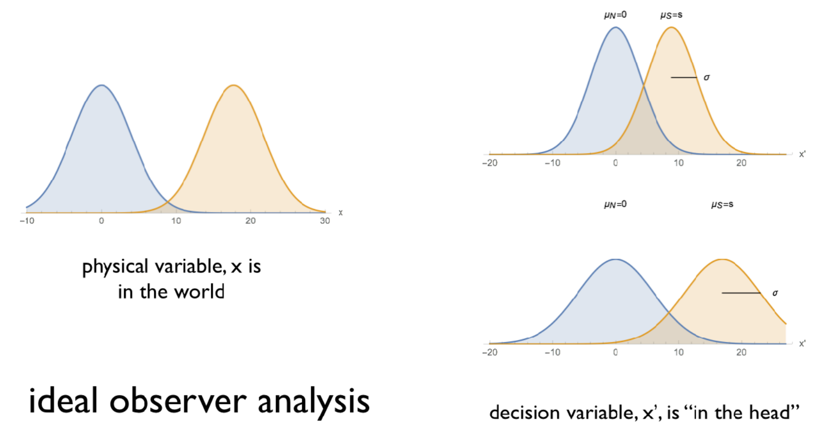
Developed signal detection theory for characterizing an ideal observer for detecting a "known" pattern in additive gaussian noise.
The statistical treatment is a special case of Bayesian inference.

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

One can study how stimuli that are objectively easily discriminable nevertheless produce human errors:



Or as in our photon/dot examples, one can study how stimuli that are objectively noisy produce human errors. The rationale for the latter approach is to distinguish external from internal sources of variability.



This time

How to study and model higher-level inference tasks with external, objective uncertainty? Extend the tools of signal detection theory to object recognition/estimation.

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.

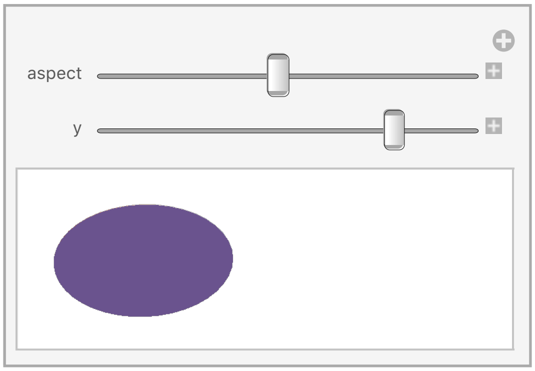
Some motivation: Examples of object tasks

Estimation

Imagine the top of a coffee mug. It typically has a circular cross-section. However, due to projection, the image on your retina is more like an ellipse from most viewpoints. Now imagine it is a "designer coffee mug" which has an elliptical cross-section. How could you guess the true, i.e. physical 3D shape, from measurements made in the projected image? The "aspect" slider below changes the ratio of the major to minor axes of the coffee mug. The "y" variable changes the slant of your viewpoint. These two causes determine an image measurement x--the height of the projected ellipse in the image (See "Slant" example below).

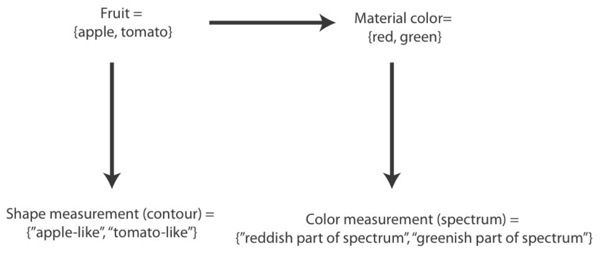
```
In[1]:= Manipulate[Graphics3D[{EdgeForm[],
           Scale[Cylinder[{{-.0, -.001, -.0}, {.0, .001, .0}}, 1 / 2], {1, 1, aspect}]},
          Boxed → False, ImageSize → 100, ViewCenter → {0, 0, 0}, ViewPoint → {0, 10, y}],
         {{aspect, 1.0}, .1, 2}, {y, -20, 20}]
```

Out[1]=



Recognition

Suppose you are doing some grocery shopping in the fruit and vegetable section. You are looking at a fruit that is either a tomato or an apple. The type of fruit will influence the regularities in measurements you have available to decide. For example, the contours of the fruit would be like what you previously experienced from apples and from tomatoes. But there is some ambiguity--silhouettes of apples aren't that different from tomatoes. Another kind of measurement could come from spectral measurements (e.g. from your cone photoreceptors), i.e. from longer-wave vs. shorter-wave parts of the spectrum. But these measurements rely on intermediate variables of material, i.e. the red or green stuff that the skin of the fruit is made of, and the color of the illumination falling on the fruit. The figure below shows the generative model. Given shape and wave-length measurements, how can one make the best guess of the fruit type and/or material type? We won't solve this problem in general, but we will look at a very simple version with a view to understanding how different kinds of tasks affect the guesses, even when the generative model remains unchanged.


Fruit =
{apple, tomato}
Material color=
{red, green}
Shape measurement (contour) =
{"apple-like", "tomato-like"}
Color measurement (spectrum) =
{"reddish part of spectrum", "greenish part of spectrum"}

Bayesian inference: general picture

Given a signal pattern, which could be a scene description S = $\{S_1,...,S_N\}$, and image measurements or features I=$\{I_1,...,I_M\}$, the "universe" of possibilities is the joint probability:

$$p(S, I)$$

“universe” in the sense that the sum over S and I sums to one. To do inference, there are two basic operations on the joint probability:

1. Condition on what is known:

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

$$p(S \mid I) = p(S, I) / p(I);$$

2. Integrate out what you don't care about

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

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

Called “summing over”, "integrating out" or "marginalization".

For example, suppose I want to calculate the ideal observer for recognizing one of 6 objects, but each object could appear in one of 12 "poses". A "pose" means a specific position relative to the camera. I'd want to set up my problem so that I can integrate over the poses, to effectively discount that source of variation. In other words I really don't want a precise estimate of the pose parameters, but I do want to be as accurate as possible in deciding which object I've seen. This kind of ideal observer analysis of human object recognition was done by Tjan et al. in 1995.

With the posterior, $p\left(S_{\text{signal}} \mid I\right)$, we can employ an optimality rule, such as “pick the signal that makes the posterior biggest for this specific measured I”. This is the MAP observer strategy we introduced earlier.

Graphical models of dependence

In the SDT example of pattern detection in noise, the generative model and task 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? Other tasks?

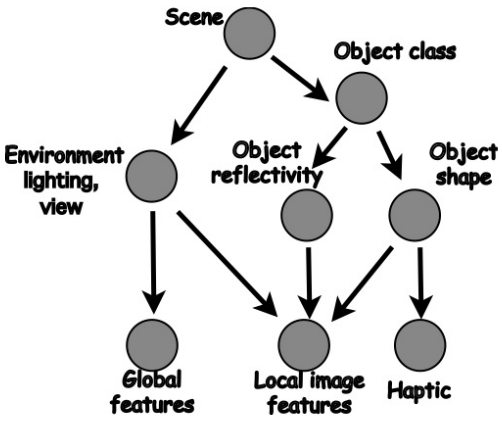
The "universe" of possible factors generating an image could be expressed by constructing the joint probability on all possible combinations of description. For example, suppose we have decided that the key variable classes to model all natural images can be broken down into descriptions of the **scene, object class, environment lighting, object reflectivity, object shape,** and that these result in several kinds of data measurements, such as **global features, local features, haptic.** Then our knowledge of the universe of natural images could be modeled as:

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 clearly hopelessly large, because of the combinatorial problem.

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 or causes might produce natural images. We can also simplify based on assumptions of what kind of information, i.e. which factors, are important to estimate.

One way to begin simplifying the problem is to note that not all variables have a direct influence on each other. Imagine you are designing a 3D software environment for quickly generating visual images, perhaps with some touch or haptic output too. We draw a graph in which lines only connect variables that influence each other. We are going to use directed graphs to represent conditional probabilities.

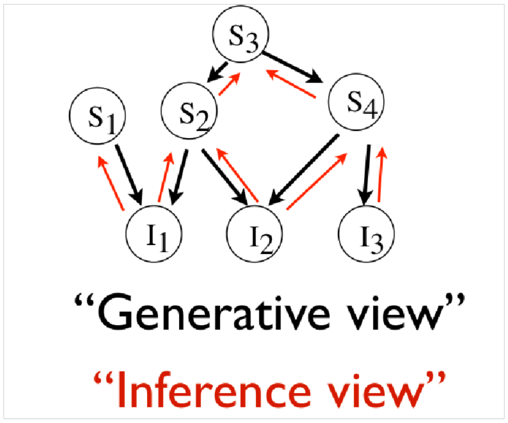


Strategy

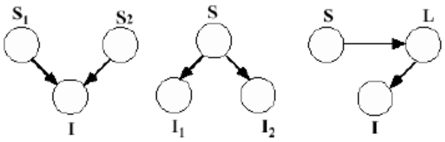
Identify the variables of interest. For perception problems, we often separate the variables into observables (I), i.e. image intensity patterns or features, and causal factors S. (We generalize this later.)

$$p(S_1,S_2,...;I_1,I_2,I_3...)$$

Use knowledge of the influence structure to simplify the joint distribution:



For example,

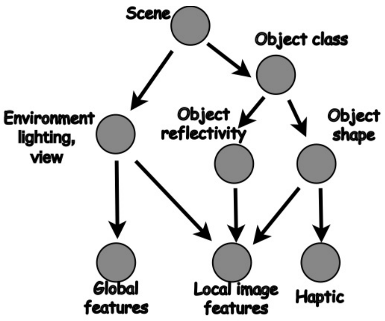


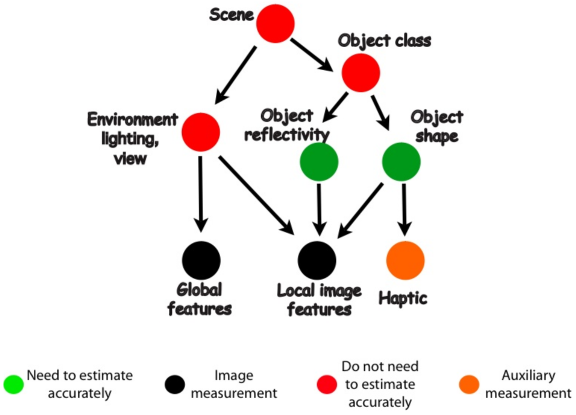
The arrows above represent a graphical shorthand that tells us how to factor a joint probability into conditionals. For the three examples above, we have:

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

$$\mathbf{p(S, I1, I2) = p(I1 \mid S)\, p(I2 \mid S)\, p(S)}$$
$$\mathbf{p(S, L, I) = p(I \mid L)\, p(L \mid S)\, p(S)}$$

Identify which variables are known, which need to be estimated, and which should be discounted.





We use a color code for nodes in our graphs: green means unknown and we'd like to estimate its value. Black means measurable or known through some means. Red marks the variables to be discounted. Sometimes, we will use orange to mean some auxiliary data caused by one of the unknown variables.

Some basic graph types in vision

To help build intuitions, let's look at the application of graphical models to the analysis of types of problems in object 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)**.

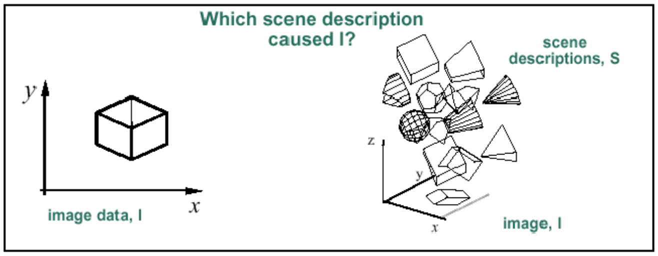
We'd like to have a specification of:

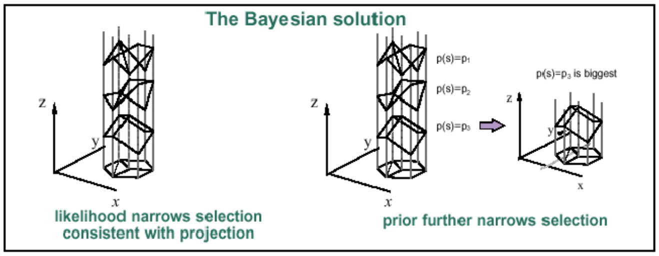
p(S|I), where is the **posterior** probability of the scene given the image

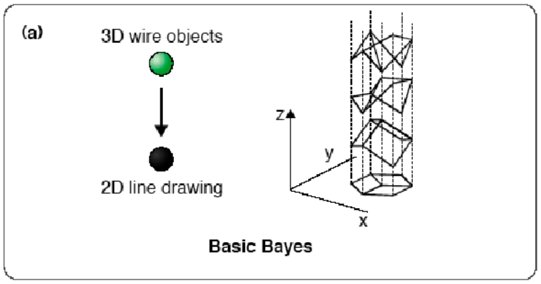
-- i.e. what you get when you condition the joint by the image data. As we’ve noted before, knowing the posterior allows us to make decisions by picking the hypothesis **S** which maximizes the posterior (i.e. maximum a posteriori or **MAP** estimation) minimizes the average probability of error.

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**.

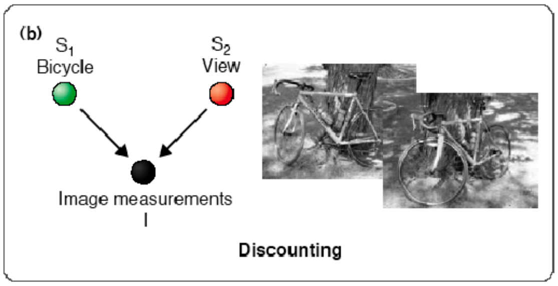



The Bayesian solution
z
y
x
likelihood narrows selection
consistent with projection
z
y
x
p(s)=p1
p(s)=p2
p(s)=p3
p(s)=p3 is biggest
z
y
x
prior further narrows selection



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.

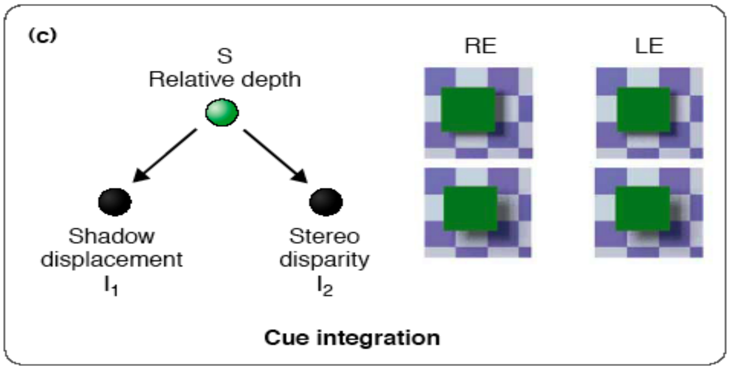
Discounting



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

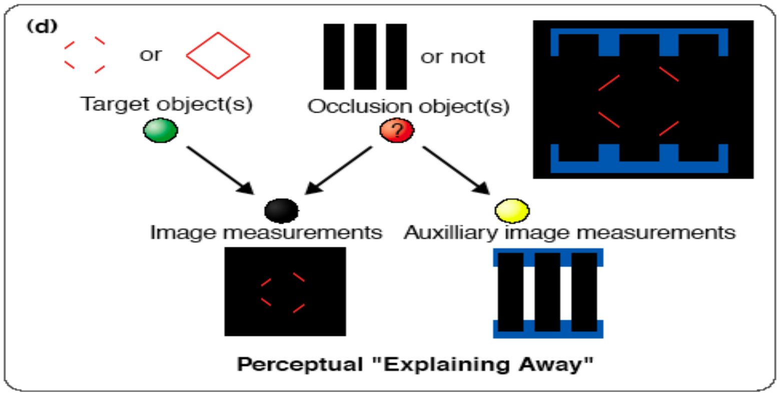
$p(S_1 \mid I)=\sum_{S_2} p(S_2, S_1 \mid I)$

Cue integration



Here two measurements (shadow displacement and stereo disparity) may be correlated. However, if S is fixed, i.e. known, 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

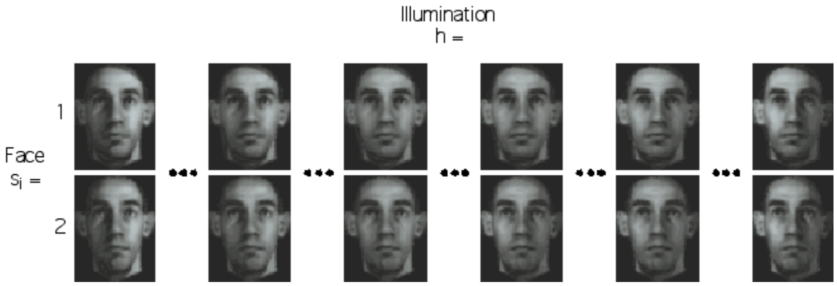
(or their flip side: utility and expected utility)

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 proportion of misses. This isn't necessarily bad.

A doctor might say that since stress EKG's have about a 30% false alarm rate in predicting heart disease, 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 also ask what the hit rate (or alternatively the miss rate) is? Miss rate is about 10%. (Thus, d' is reasonably high--what is it?). From the patient's point of view, the cost of a miss is very high, possibly 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 personal cost that puts a very high cost on a miss, and lower cost on a false alarms.

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



One example is in shape from shading. An image provides the "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 on task. The effect of marginalization in the fruit example below illustrates the effect of task-dependence on optimal decisions.

To get insight into the ideas behind decision theory, let’s look at an example where the costs of errors are continuous functions of the estimates of object properties.

Slant estimation example using Bayes decision theory

Recall the coffee mug top example at the beginning of the lecture. We are given one simple measurement x -- the height of the ellipse in the image. From it, we'd like to estimate the most probable values of the shape of the ellipse and the viewpoint. We assume a fixed width (say unit 1), so the shape is characterized by the aspect ratio--i.e. the physical height of the top in 3D. And the viewpoint is characterized by the slant, as shown in the figure below.

So given one number x (e.g. x = 0.5), we have to estimate two unknowns about the physical state of affairs:

-- slant of the top with respect to the viewer and

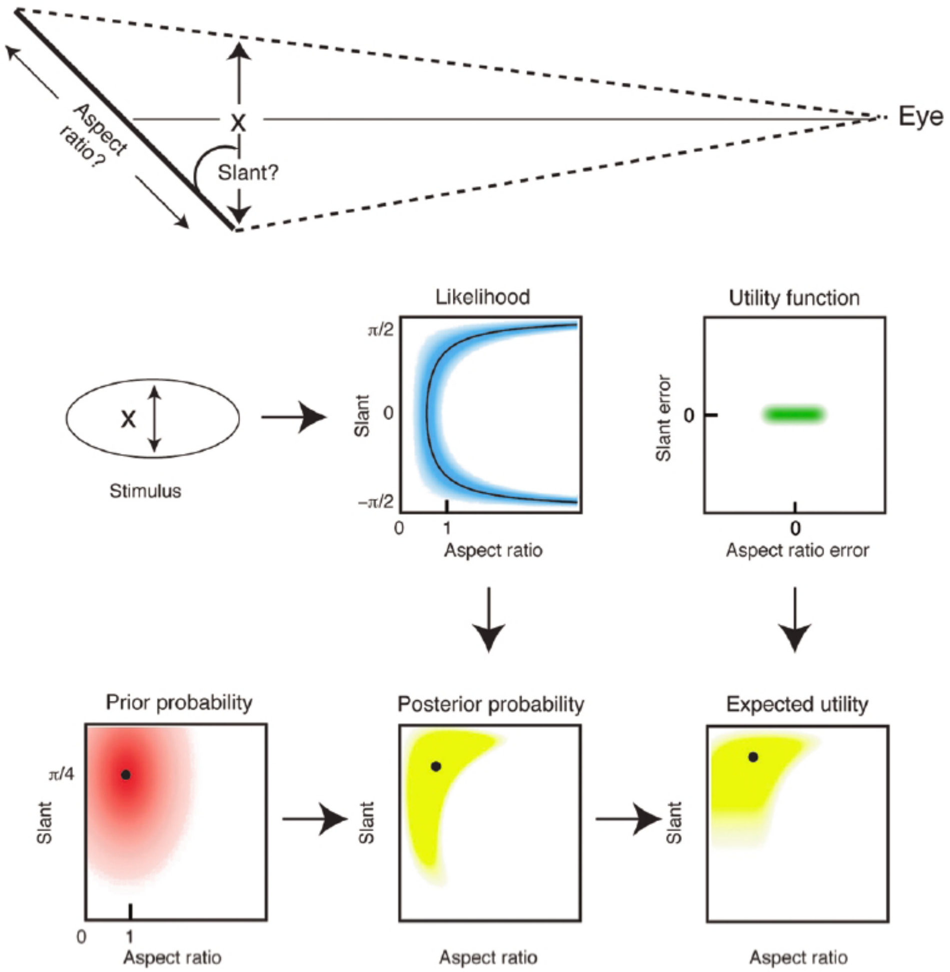
-- aspect ratio of the top of the cup, which in our simplified case is just the physical height of the disk.

We will use three types of constraints:

--a generative model: x = aspectratio*Cos[slant] + noise

--prior assumptions about typical shapes and viewpoints

--assumptions about what is more important to get right, the slant or the aspect ratio



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

Introduction

Consider the above figure which illustrates a Bayesian ideal observer for a tasks involving the perception of object properties that differ along two physical dimensions, such as aspect ratio and slant (other examples are size and distance, or speed and direction of motion).

When a stimulus is received, in this case a measurement x = 0.5, 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. The peak of this gives the most probable estimate, but not necessarily the most useful.

So 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 = 1/2) 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.

Initialization

```
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}];*)
```

Calculate Likelihood function and its maxima

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

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

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

The generative or 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)

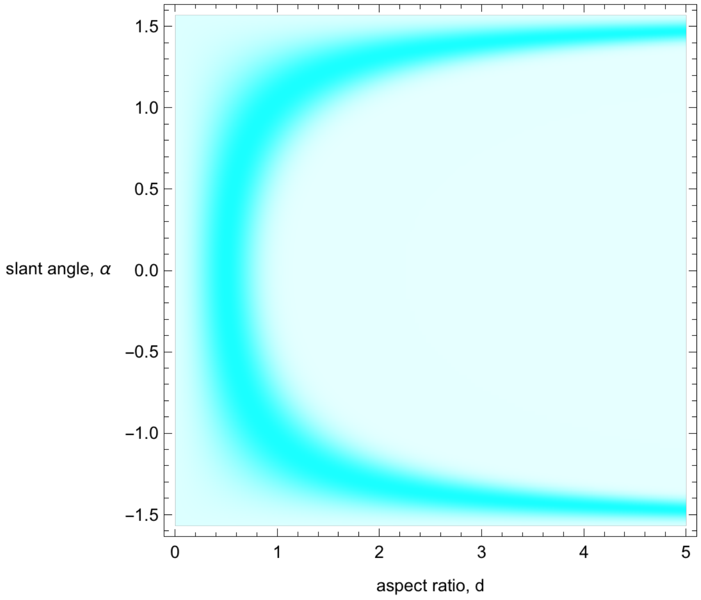
```
likeli[alpha_, x_, d_, s_] :=
 Exp[-((x - d Cos[alpha])^2) / (2 s^2)] (1 / Sqrt[2 Pi s^2])
likeli[α, x, d, s]
x = 1 / 2; s = 1 / 5;
like = likeli[α, x, d, s]
```

$$\frac{e^{-\frac{(x-d\,\mathrm{Cos}[\alpha])^2}{2\,s^2}}}{\sqrt{2\,\pi}\,\sqrt{s^2}}$$

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

Plot likelihood

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



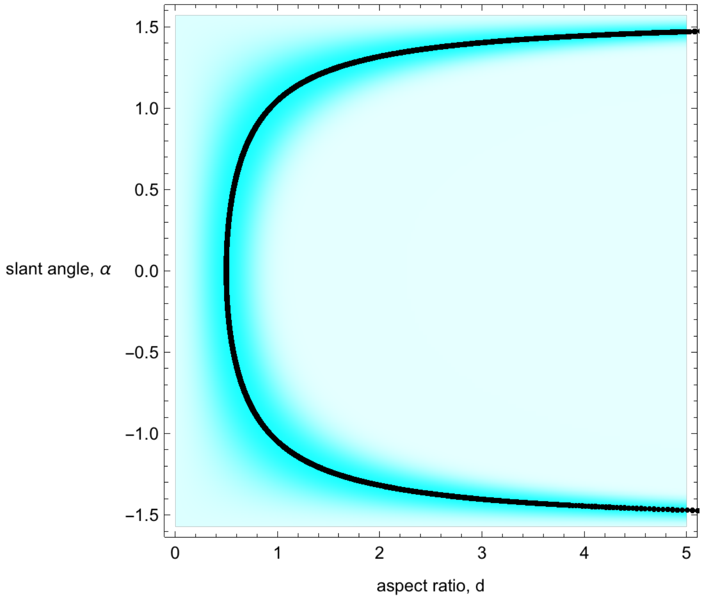
Plot likelihood maxima

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

```
temp2 = Table[Point[{x/Cos[alpha], alpha}], {alpha, -π/2, π/2, 0.001}];

temp = Join[Table[Point[{d, ArcCos[x/d]}], {d, loaspect + 0.5, hiaspect, 0.01}], temp2];

gtemp = Graphics[{PointSize[0.01], temp}];
```

Plot likelihood together with maximum along the ridge

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

glikemax = Show[gdlike, gtemp]
```



Calculate the prior, and find its maximum

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

$$p(\alpha, d)$$

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:

```
Clear[R, x, y];
```

$$R = \{\{\sigma_x^2, \rho\,\sigma_x\,\sigma_y\}, \{\rho\,\sigma_x\,\sigma_y, \sigma_y^2\}\};\ \texttt{(*}\rho\texttt{ is the correlation between }\alpha\texttt{ and d*)}$$

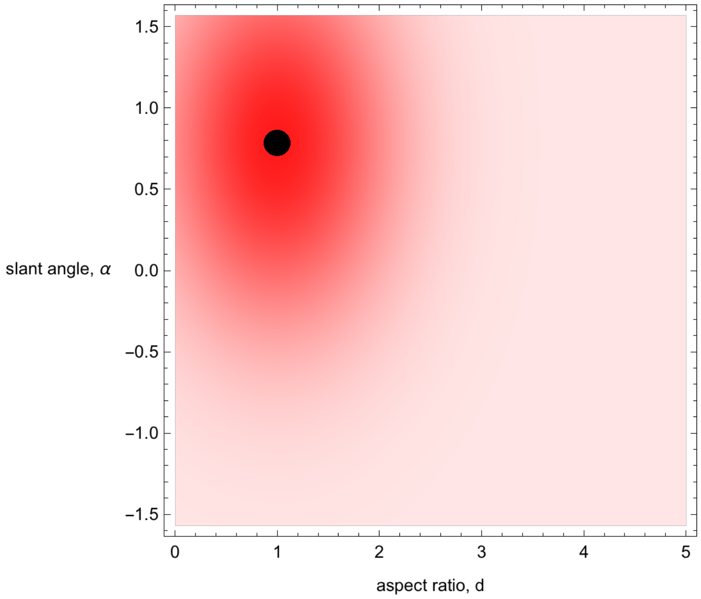
$$\texttt{PDF}\left[\texttt{MultinormalDistribution}\left[\{\mu_x, \mu_y\}, R\right], \{x, y\}\right]$$

$$e^{\frac{1}{2}\left(-\frac{(x-\mu_x)\,(y\,\rho\,\sigma_x-\rho\,\mu_y\,\sigma_x-x\,\sigma_y+\mu_x\,\sigma_y)}{(-1+\rho^2)\,\sigma_x^2\,\sigma_y}-\frac{(y-\mu_y)\,(-y\,\sigma_x+\mu_y\,\sigma_x+x\,\rho\,\sigma_y-\rho\,\mu_x\,\sigma_y)}{(-1+\rho^2)\,\sigma_x\,\sigma_y^2}\right)} \Bigg/ \left(2\,\pi\,\sqrt{\sigma_x^2\,\sigma_y^2-\rho^2\,\sigma_x^2\,\sigma_y^2}\right)$$

```
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];
```

$$\{-7.35283\times 10^{-15}, \{d \to 5., \alpha \to 1.\}\}$$

```
Show[gdprior, Graphics[{PointSize[0.05`], Point[{1, 0.785`}]}]]
```



Calculate the posterior, and find its maximum

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

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

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

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];
```

$$\texttt{R} = \{\{\sigma_\alpha^2, \rho\,\sigma_\alpha\,\sigma_d\}, \{\rho\,\sigma_\alpha\,\sigma_d, \sigma_d^2\}\};\ \texttt{(*}\rho\texttt{ is the correlation between }\alpha\texttt{ and d*)}$$

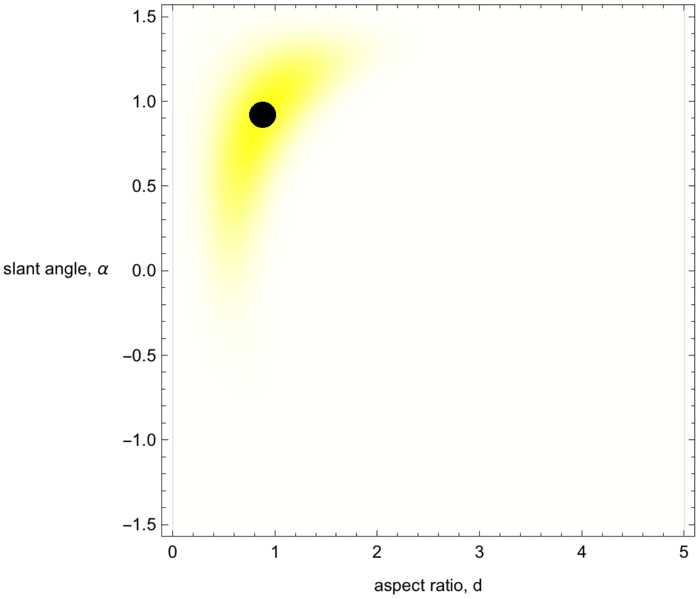
$$\texttt{likeli[}\alpha\texttt{, x, d, s] * PDF[MultinormalDistribution[\{}\mu_\alpha\texttt{, }\mu_d\texttt{\}, R1], \{}\alpha\texttt{, d\}]}$$

$$\frac{1}{\sqrt{s^2}}\,0.253975\,e^{-\frac{(x-d\,\mathrm{Cos}[\alpha])^2}{2\,s^2}+\frac{1}{2}\,(-(0.+4.\,(d-\mu_d))\,(d-\mu_d)-(0.+4.\,(\alpha-\mu_\alpha))\,(\alpha-\mu_\alpha))}$$

```
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, PlotRange → {0, 130}];
FindMinimum[-pdf3 * like, {d, 1.0}, {α, 0.2}]
```

```
{-1.17378, {d → 0.881475, α → 0.923647}}
```

```
Show[gdpost, Graphics[{PointSize[0.05], Point[{0.88, 0.92}]}]]
```



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

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

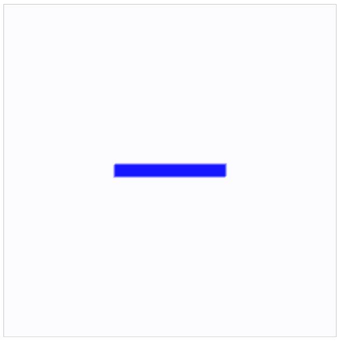
risk=posterior*loss,   where * means convolve; utility = -risk

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, 0.25]) (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), 1 - (0.01 + 0.9 #1), 1] &),
   Frame → False, ImageSize → Small]
```



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

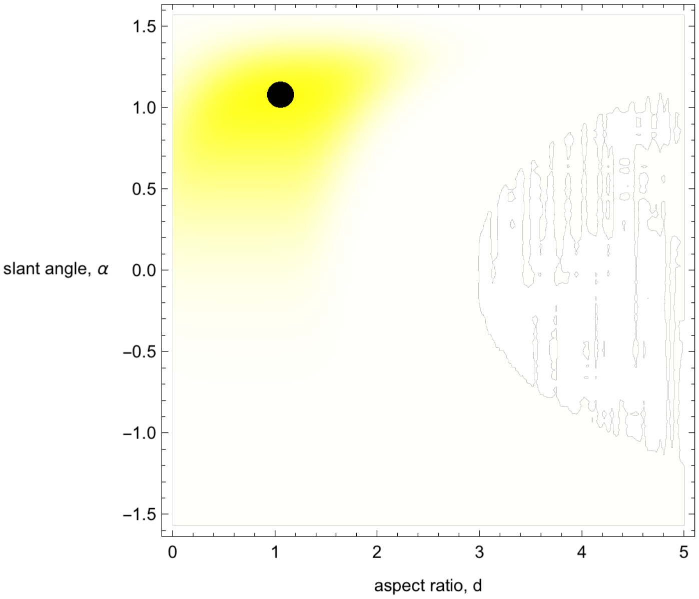
```
post = Transpose[Table[like * pdf3, {d, loaspect, hiaspect,
      (hiaspect - loaspect) / npoints}, {α, -Pi / 2, Pi / 2, Pi / npoints}]];
post2 = PadMatrix[post, 0, 16];
maploss2 = PadMatrix[maploss, 0, 16];
offset = Floor[Dimensions[maploss2][[1]] / 2];
tempcon = ListConvolve[maploss2, post2, {-1, -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, PlotRange → {0, 130},
   DataRange → {{loaspect, hiaspect}, {-Pi / 2, Pi / 2}}];
```

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

```
{{108, 27}}
```

```
Show[grbrisk, Graphics[
  {PointSize[0.05`], Point[{27. * 5 / npoints, (108. - 64) * (Pi / (npoints))}]}]]
```



▸ 1. Exercise: Compute and plot expected loss for the transposed loss function, i.e. where precision in the aspect ratio is more important than precision in slant.

Bayes decision theory: Formal summary

Formalization of decision theory: loss and risk

Here we formalize the intuitions provided by the above demonstration to 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 probability and 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 distinguish the decision (or action) 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:

$$R(d;x) = \sum_s L(d,s)\, p(s \mid x)$$

This suggests a decision rule: $\alpha(x)=\underset{d}{\text{argmin}}\, R(d;x)$. But not all x are equally likely. In principle, we should pick our decision rule to minimize the expected risk averaged over all observations:

$$R(\alpha) = \sum_x R(d;x)\, p(x)$$

Maximum likelihood, MAP, and marginalization are special cases of the choice of loss function

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,

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

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), i.e. finding the “mode” of the posterior.

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

L = -1.

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

So for our face recognition example, a really huge error in illumination direction would have the same cost as getting it right. For the fruit example, below, optimal classification of the fruit identity requires marginalizing over fruit color--i.e. effectively treating fruit color identification errors as equally costly.

▶ 2. What loss function corresponds to estimating the mean rather than the mode?

Inference: Fruit classification example

This section provides another quantitative example of visual inference, where we use a graphical model. It illustrates how the task choice, and therefore what you integrate out, can change what is the optimal decision.

For intuitions and background on the relation between graphical models and conditional dependence, see the Supplementary Reading: Griffiths, T. L., & Yuille, A. (2008).

Here we assume MAP estimation, but over posteriors differing by what we decide to integrate out. The utility function is implicit--all errors in the secondary variables are equally costly.

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

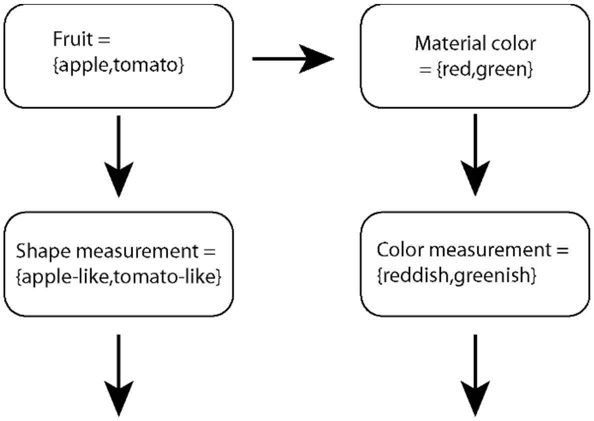


Figure from Yuille, Coughlan, Kersten & Schrater.

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]

Let’s see how to go from a graph to a factorization of the joint distribution. Note that we go from left to right on the right hand side of the above equation. The idea is to find the top node (parent) which defines the prior p(F), and then look for its children, which define the conditionals:

p(Is|F) and p(C|F).

The “C” node is a child of F, but also a parent to the color measurement, so the last factor is p(Ic|C), (again remember we are going from right to left).

Below we apply basic rules of probability:

p(X,Y) = p(X|Y)p(Y), the product rule, which implies conditioning: p(X|Y) = p(X,Y)/p(Y) p(X) = $\sum_Y p(X, Y)$, the sum rule for marginalization.

The prior model on hypotheses, F & C

More apples (F=1) than tomatoes (F=2) on average, 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	$\frac{7}{16}$

Assume the conditional probability **cpCF[C|F] is**:

```
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:

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

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)=$\sum_C \sum_F p(F, C, \mathrm{Is}, \mathrm{Ic})$, 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) = $\frac{p(F,C,\mathrm{Is},\mathrm{Ic})}{p(\mathrm{Is},\mathrm{Ic})} \propto$ 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, 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"

In this case, we want maximize the posterior:

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

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

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

$$\begin{array}{l|l} C=r & \frac{115}{157} \\ C=g & \frac{42}{157} \end{array}$$

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

```
{{1}}
```

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

Pick most probable fruit--Answer "apple"

p(F | Is, Ic)

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

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

$$\begin{array}{l|l} F=a & \frac{803}{1413} \\ F=t & \frac{610}{1413} \end{array}$$

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

```
{{1}}
```

The answer is "apple"

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

Exercises

▶ 3. 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:

$$p(\text{error} \mid x) = p(\text{say "bright"}, \text{ actually dim} \mid x) + p(\text{say "dim"}, \text{ actually bright} \mid x) =$$
$$p(\hat{s}1, \ s2 \mid x) + p(s1, \ \hat{s}2 \mid x)$$

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.

4. 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,

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

We want to show that: p(error) - p(error*)≥0.

5. 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

SDT in a graphical nutshell

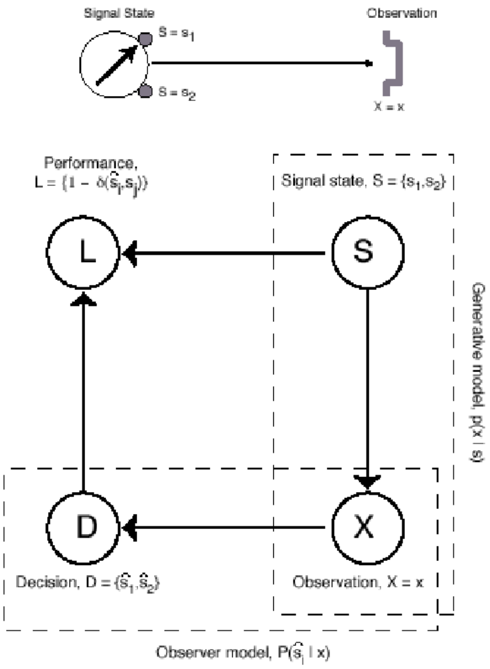
This section shows the common structure shared by three types of inference: detection, classification, and estimation.

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.

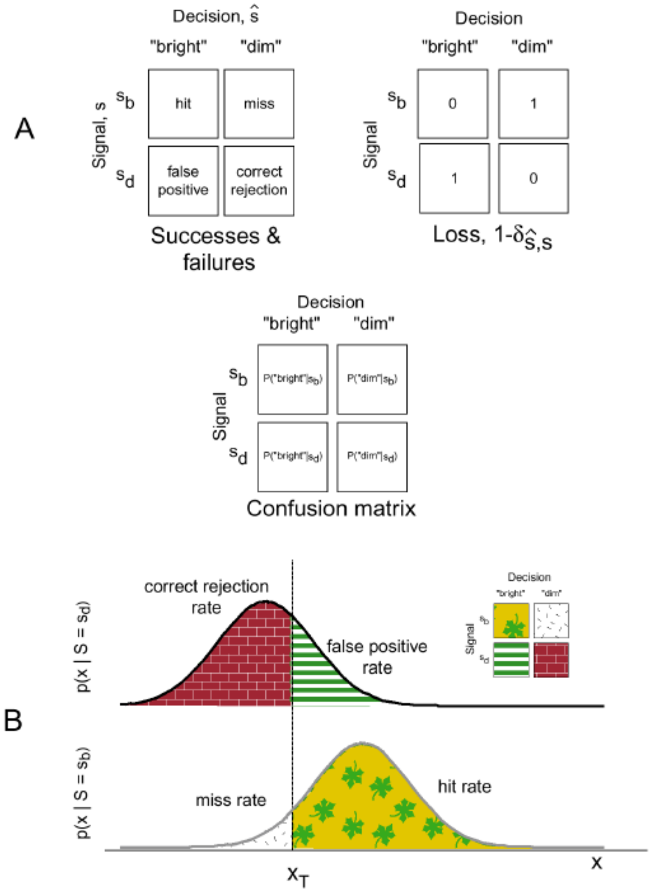
Hypothesis inference: Three types

Detection

Let the hypothesis variable d, be represented by $\hat{s}$ which can take on two values.

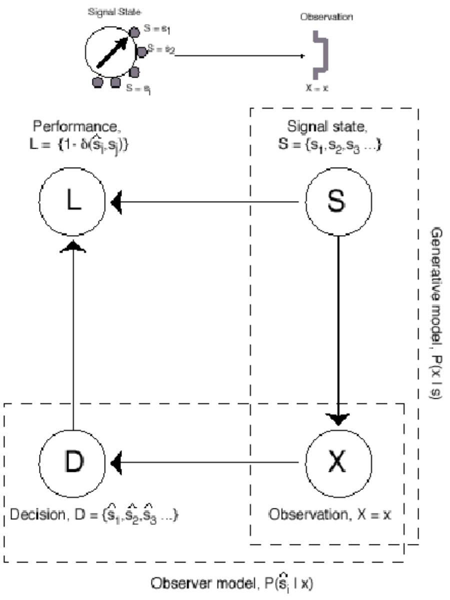

Signal State
$S = s_1$
$S = s_2$
Observation
$X = x$
Performance,
$L = \{1 - \delta(\hat{s}_i, s_j)\}$
Signal state, $S = \{s_1, s_2\}$
L
S
D
X
Decision, $D = \{\hat{s}_1, \hat{s}_2\}$
Observation, $X = x$
Generative model, $p(x \mid s)$
Observer model, $P(\hat{s}_i \mid x)$

loss function for yes/no task



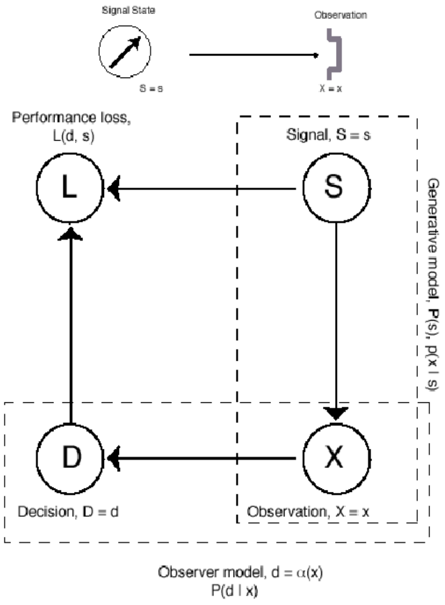
Classification

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



Continuous 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).

Graphical models for inference, synthesis, learning

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.

As above, we will use a color code for nodes in our graphs: green means unknown and we'd like to estimate its value. Black means "known" (either through a measurement or other source of knowledge).

Three main types of inference

In general, we are interested in how images are generated, how we can inference the causes of images, and how can we learn the relationship between causes and images. All three problems can be treated as decisions about, given the joint probability of image (I), causes (S) and intermediate variables (L), what to condition on and what to integrate out.

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 for example, 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.)

Learning (parameter inference)

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.
(E.g. 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.

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.

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

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

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

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

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

```
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
```

$$\left\{\left\{\frac{1}{4}, \frac{1}{4}, \frac{1}{2}\right\}, \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right\}\right\}$$

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]
```

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

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]]]
```

BarChart: Options expected (instead of $\{\frac{1}{3}, \frac{1}{6}, \frac{1}{6}\}$) beyond position 1 in BarChart$[\{\frac{1}{12}, \frac{1}{12}, \frac{1}{6}\}, \{\frac{1}{3}, \frac{1}{6}, \frac{1}{6}\}]$. An option must be a rule or a list of rules.

$$\text{BarChart}\left[\left\{\frac{1}{12}, \frac{1}{12}, \frac{1}{6}\right\}, \left\{\frac{1}{3}, \frac{1}{6}, \frac{1}{6}\right\}\right]$$

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}]
```

```
{{m[1, 1], m[1, 2], m[1, 3]}, {m[2, 1], m[2, 2], m[2, 3]}}
```

```
{x[1], x[2]}
```

```
M/X
```

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

Putting the probabilities back together again to get the joint

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

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

```
pxH pH
```

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

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 generativemodel 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)]
```

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

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}}
```

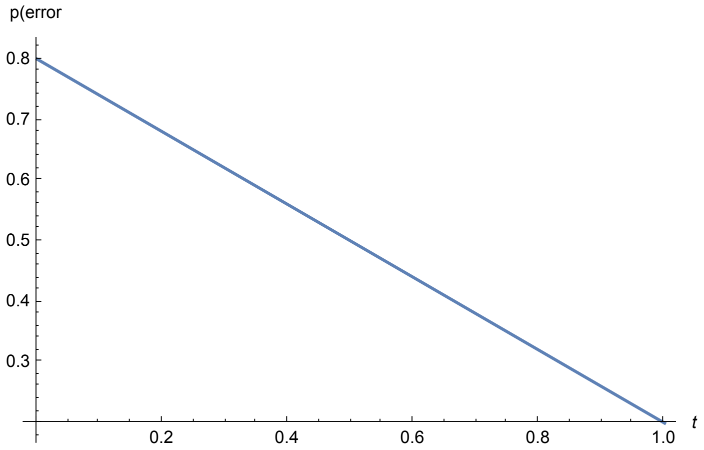
$$\left\{\left\{\frac{1}{8}, \frac{1}{8}, \frac{1}{4}\right\}, \left\{\frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}\right\}$$

Figure code

Projection ambiguity and the estimation of pose (slant) and object dimensions (aspect ratio).

```
GraphicsRow[{Graphics3D[{EdgeForm[], Cylinder[{{0, 0, 0}, {.01, .01, .001}}, 1]},
   ImageSize → Tiny, Boxed → False, AspectRatio → 1],
  Graphics3D[{EdgeForm[], Cylinder[{{0, 0, 0}, {.01, .01, .001}}, 1]},
   ImageSize → Tiny, Boxed → False, AspectRatio → 1 / 2]}]
```

```
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.

Tjan, B. S., Braje, W. L., Legge, G. E., & Kersten, D. (1995). Human efficiency for recognizing 3-D objects in luminance noise. Vision Res, 35(21), 3053-3069.

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)