# Introduction to Neural Networks
# U. Minn. Psy 5038

# Unifying neural network computations using Bayesian decision theory

### ■ Initialize:

```
In[388]:=  << "BarCharts`";
           << "Histograms`"
           << "MultivariateStatistics`"
```

```
In[391]:=  Off[General::spell1];
```

```
In[392]:=  SetOptions[ListDensityPlot, ImageSize → Small];
           SetOptions[DensityPlot, ImageSize → Small];
           SetOptions[ContourPlot, ImageSize → Small];
```

## Bayesian Decision Theory: Utility

### Review: Graphical Models of dependence

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

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.

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

#### ■ Condition on what is known:

Given a state of the world S, and inputs I, the "universe" of possibilities is:

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

If we know I (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 we don't care about

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

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

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 input 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 input data I itself. Consider three very broad types of task:

### ■ Data inference: synthesis

**Data synthesis** (generative or forward 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 or estimation

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 identification. Again there is an intermediate variable, L, it gets integrated out.

### Learning (parameter inference)

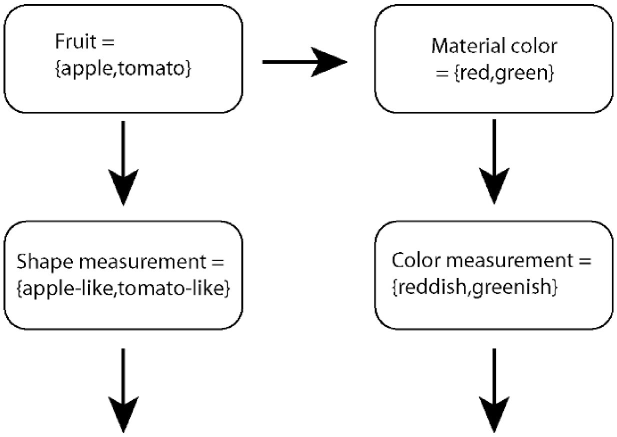
**learning** can also be viewed as estimation**:** we want to model L: p(L|I,S), to learn how the intermediate variables are distributed. Given lots of samples of outputs and their inputs, we want to learn the mapping parameters between them. (Alternatively, do a mental switch and consider a neural network 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.

## Recall: Fruit classification example



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 ]

### ■ Three MAP tasks

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

Pick most probable color--Answer "red"

Pick most probable fruit--Answer "apple"

## Some basic graph types in vision

### Basic Bayes

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

**S** (the scene), and **I** is (the image data), and **I = f(S)**.

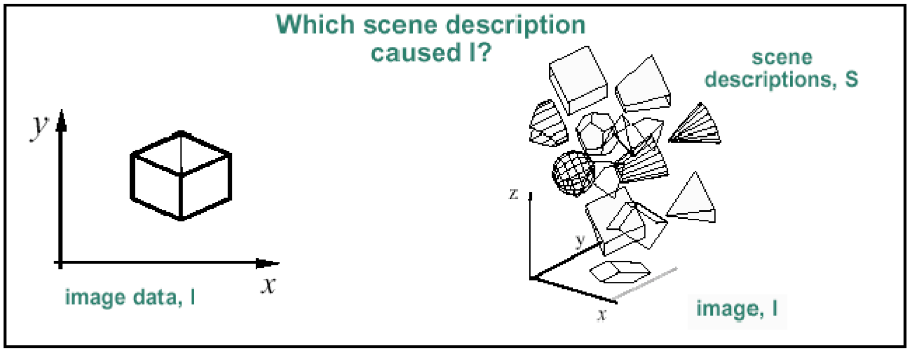
We'd like to have:

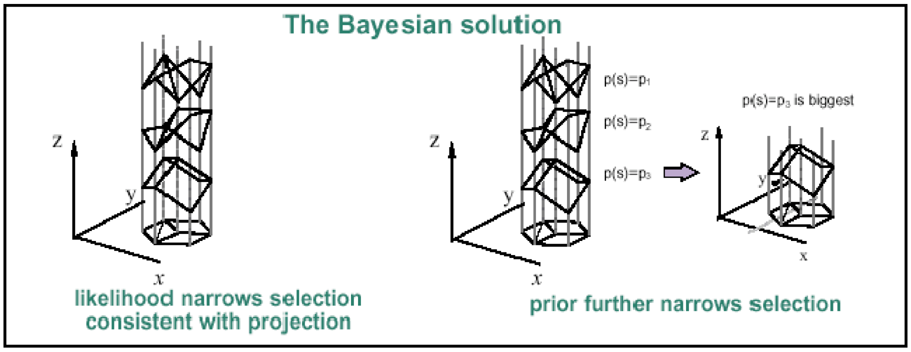
**p(S|I)**, where is the **posterior** probability of the scene given the image

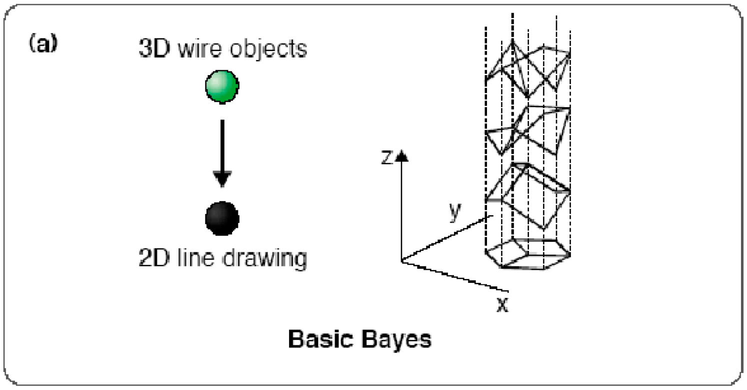
-- i.e. what you get when you condition the joint by the image data. The posterior is often what we'd like to base our decisions on, because as we discuss below,  picking the hypothesis **S** which maximizes the posterior (i.e. maximum a posteriori or **MAP** estimation) minimizes the average probability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

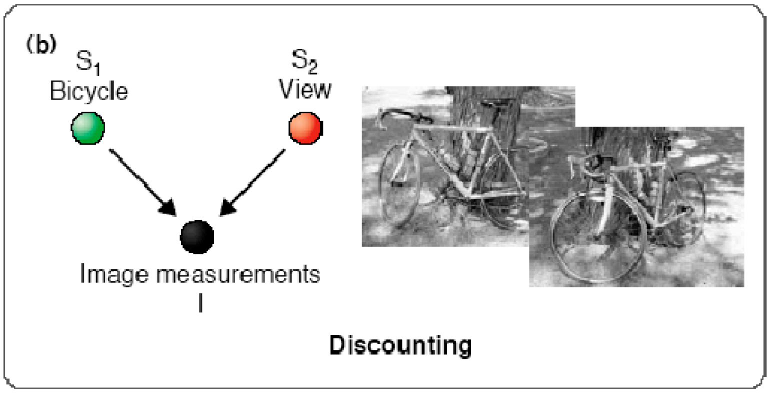



The Bayesian solution
z
y
x
likelihood narrows selection consistent with projection
z
y
x
p(s)=p1
p(s)=p2
p(s)=p3
p(s)=p3 is biggest
z
y
x
prior further narrows selection



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.

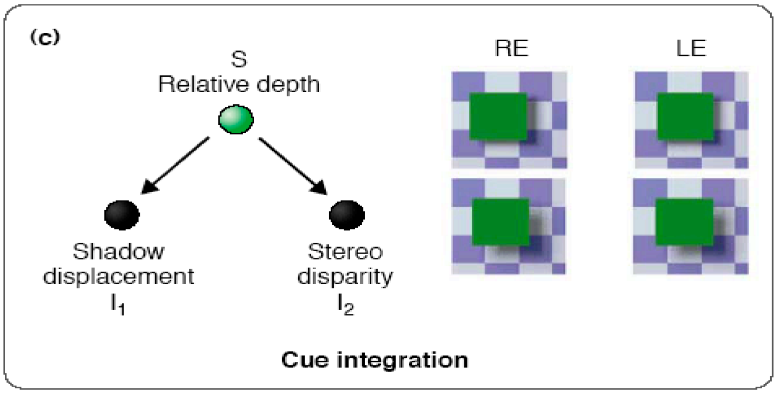
## Discounting



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

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

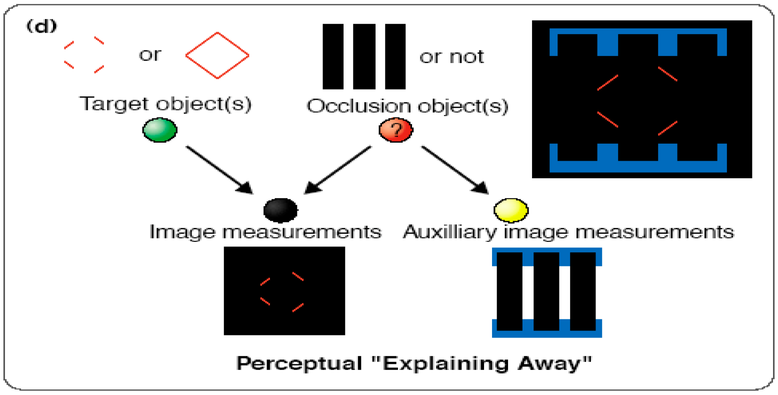
## Cue integration



Optimal cue weighting for gaussian case.

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

### Explaining away



*How to generalize optimal inference to include task requirements?*

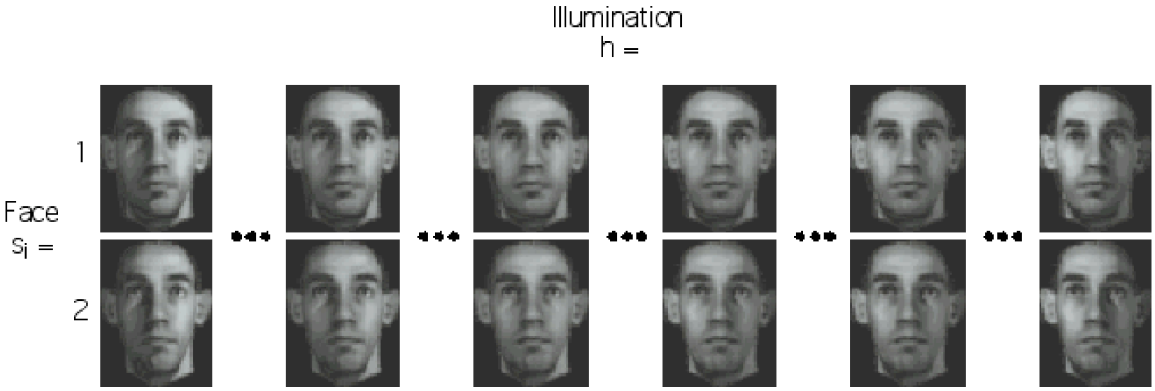
## Bayes Decision theory, loss, and risk

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.

The costs of certain kinds of errors (e.g. a high cost to false alarms) can affect the decision criterion. Even though the sensitivity of the observer is essentially unchanged (e.g. the d' = $(\mu_2 - \mu_2)/\sigma$ for 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%. 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, the concept of utility is relevant even in perception. Perception has implicit, unconscious trade-offs in the kinds of errors that are made.



For example, image intensities provide  the data 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. The terms state, hypothesis, signal state as essentially the same--to represent the random variable indicating the state of the world--the "state space". We often assume 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:

$$R\ (d;\ x)\ =\sum_{s} L\ (d,\ s)\ p\ (s\ |\ x) \tag{4}$$

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

$$R\ (\alpha)\ =\sum_{x} R\ (d;\ x)\ p\ (x) \tag{5}$$

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,

$$R\ (d;\ x)\ =\sum_{s} L\ (d,\ s)\ p\ (s\ |\ x)\ =\sum_{s}\left(1-\delta_{d,s}\right)\ p\ (s\ |\ x)\ =1-p\ (d\ |\ x) \tag{6}$$

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 = $\left(1 - \delta_{d,s}\right)$ 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.

$$R(s1; x) = \sum_{s2} L(d2, s2)\, p(s1, s2 \mid x) \tag{7}$$

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.

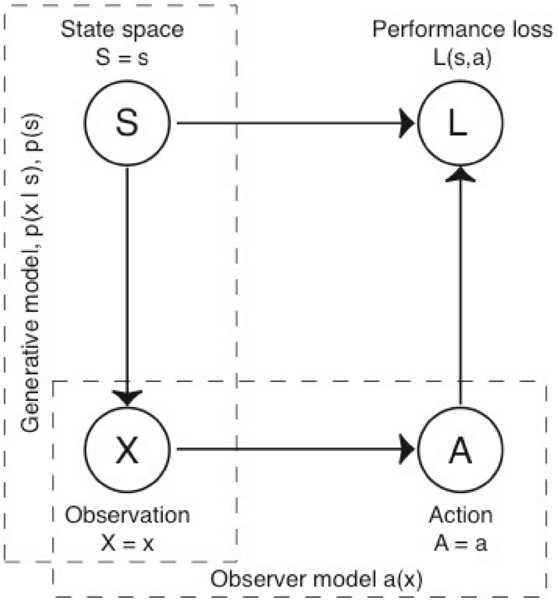
## Graphical model for decision theory

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.

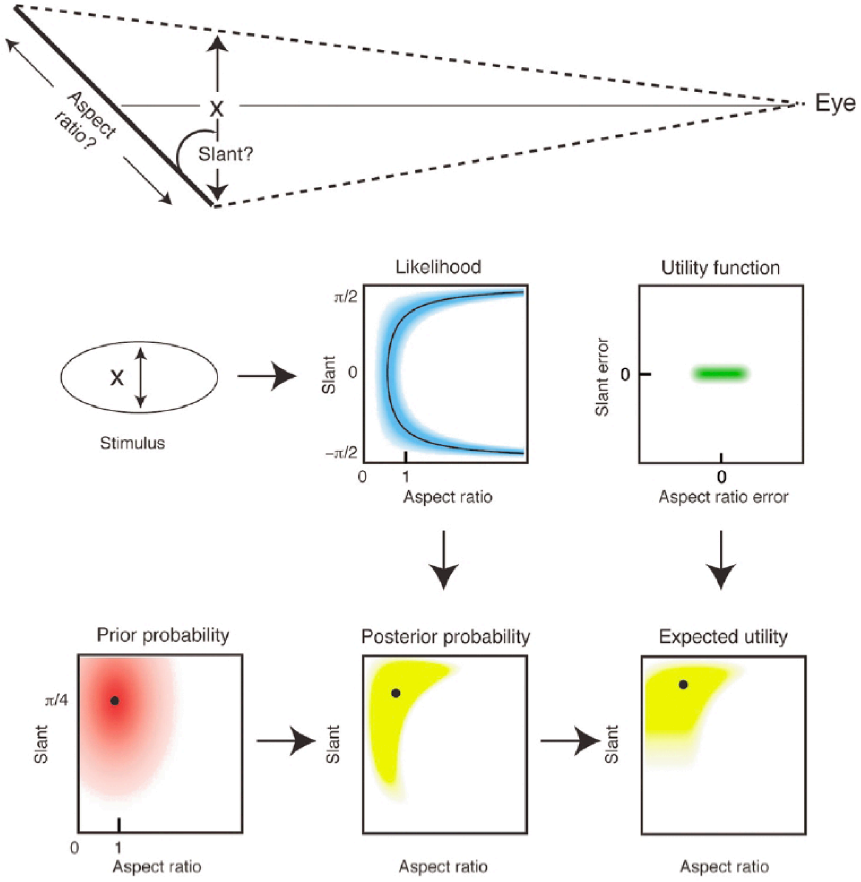
In the diagram below, we replace the decision variable d, by a more general term a for "action".

## Decisions


State space
S = s
Performance loss
L(s,a)
S
L
Generative model, p(x | s), p(s)
X
A
Observation
X = x
Action
A = a
Observer model a(x)

- **Detection: a = estimate of s = s' = $\{s'_1, s'_2\}$, e.g. s = $\{s'_1$, not $s'_1\}$**
- **Classification: a = s' = $\{s'_1, s'_2, s'_3, s'_4, ...\}$**
- **Estimation: a = s', where s' takes on continuous values**

## 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

In[395]:=

```
<< "BarCharts`"; << "Histograms`"
<< "MultivariateStatistics`"
```

In[397]:=

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

Out[400]=

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

In[402]:=

```
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

In[403]:=

```
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)

In[404]:=

```
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]
```

Out[405]=

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

Out[407]=

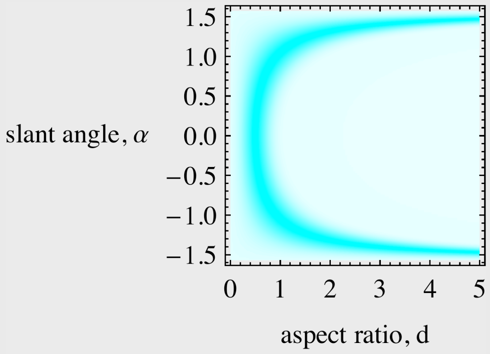
$$\frac{5\,e^{-\frac{25}{2}\left(\frac{1}{2}-d\cos(\alpha)\right)^2}}{\sqrt{2\pi}}$$

## Plot likelihood

In[408]:=

```
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]
```

Out[408]=



## Plot likelihood maxima

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

In[409]:=

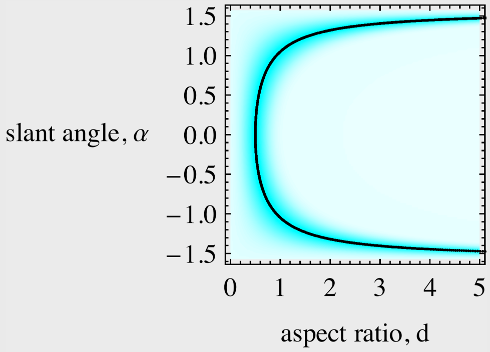
```
temp2 = Table[Point[{x/Cos[alpha], alpha}], {alpha, -π/2, π/2, 0.001`}];
temp =
 Join[Table[Point[{d, ArcCos[x/d]}], {d, loaspect + 0.5`, hiaspect, 0.01`}],
  temp2];
gtemp = Graphics[{PointSize[0.01`], temp}];
```

## Plot likelihood together with maximum along the ridge

In[411]:=

```
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]
```

Out[412]=



- **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:

In[413]:=

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

Out[413]=

PDF[MultinormalDistribution($\{\mu_\alpha, \mu_d\}$, $R$), $\{\alpha, d\}$]

In[414]:=

```
R1 = {{.25, 0}, {0, .25}};
ndist3 = MultinormalDistribution[{Pi / 4., 1}, R1];
pdf3 = PDF[ndist3, {α, d}];
FindMinimum[-pdf3, {{d, 0}, {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];
```

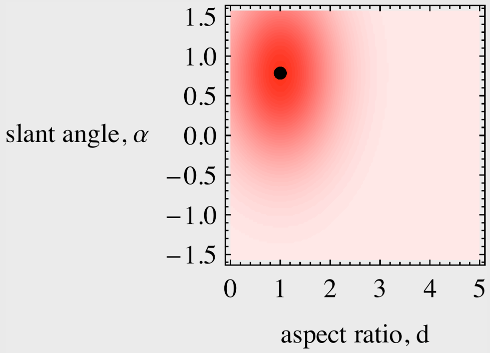
Out[417]=

$\{-0.63662, \{d \to 1., \alpha \to 0.785398\}\}$

In[419]:=

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

Out[419]=



- **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:

In[420]:=

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

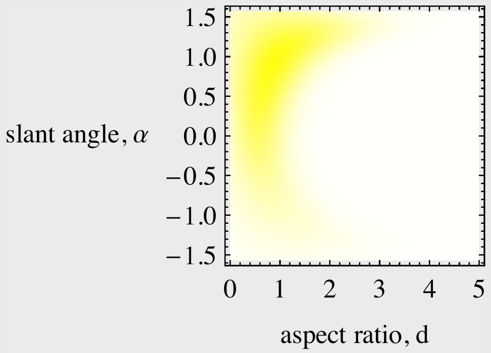
Out[421]=

$$\frac{e^{-\frac{(x-d\cos(\alpha))^2}{2s^2}}\ \text{PDF}[\text{MultinormalDistribution}(\{\mu_\alpha, \mu_d\}, R), \{\alpha, d\}]}{\sqrt{2\pi}\ \sqrt{s^2}}$$

In[422]:=

```
gdpost = DensityPlot[(pdf3 * like) ^ .2, {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]
```

Out[422]=



In[423]:=

```
R1 = {{.25, 0}, {0, .25}};
ndist3 = MultinormalDistribution[{Pi / 4., 1}, R1];
pdf3 = PDF[ndist3, {α, d}]
FindMinimum[-pdf3 * like, {{d, 1}, {α, 1}}]
```

Out[425]=

$$0.63662\, e^{\frac{1}{2}(-(d-1)\,(4.\,(d-1)+0.\,(\alpha-0.785398))-(0.\,(d-1)+4.\,(\alpha-0.785398))\,(\alpha-0.785398))}$$

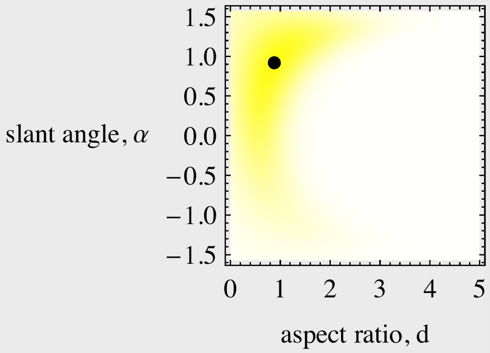
Out[426]=

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

In[427]:=

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

Out[427]=



### ■ 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

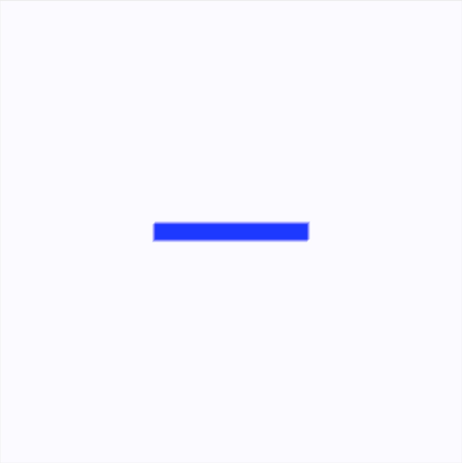
$$l(\Delta\alpha, \Delta d) = l(\alpha' - \alpha, d' - d)$$

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?).

In[428]:=

```
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]
```

Out[429]=



## 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

In[430]:=

```
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}];
```

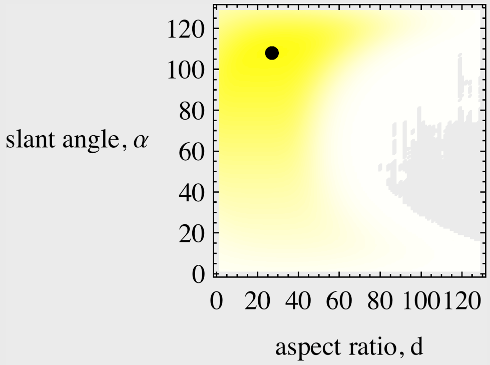
```
In[437]:= grbrisk = ListDensityPlot[Map[#^1. &, risk]^.2, Mesh → False,
              ColorFunction -> (RGBColor[1, 1, 1 - (0.01 + 0.9 #)] &),
              FrameLabel → {"aspect ratio, d", "slant angle, α"},
              RotateLabel → False, Frame → Fswitch];
```

```
In[438]:= Position[(risk), Max[(risk)]]
```

```
Out[438]= ( 108   27 )
```

```
In[439]:= Show[grbrisk, Graphics[{PointSize[0.05`], Point[{27, 108}]}]]
```

Out[439]=



## How is utility represented by the brain?

Natural loss functions may be "hard-wired", embedded in the architecture.

But dynamic changes? The role of reward.

# References

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

Glimcher, P. (2002). Decisions, decisions, decisions: choosing a biological science of choice. *Neuron, 36*(2), 323-332.

Glimcher, P. W., & Rustichini, A. (2004). Neuroeconomics: the consilience of brain and decision. *Science, 306*(5695), 447-452.

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

Hsu, M., Bhatt, M., Adolphs, R., Tranel, D., & Camerer, C. F. (2005). Neural systems responding to degrees of uncertainty in human decision-making. *Science, 310*(5754), 1680-1683.

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)