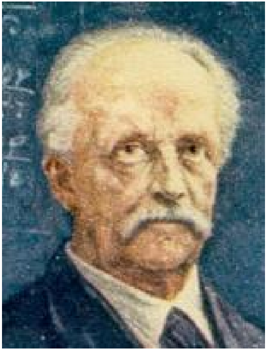
Coding (and computing with) Uncertainty

Maneesh Sahani

Gatsby Computational Neuroscience Unit
University College London

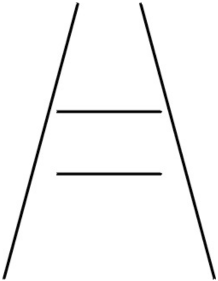
March 2017

Helmholtzian inference



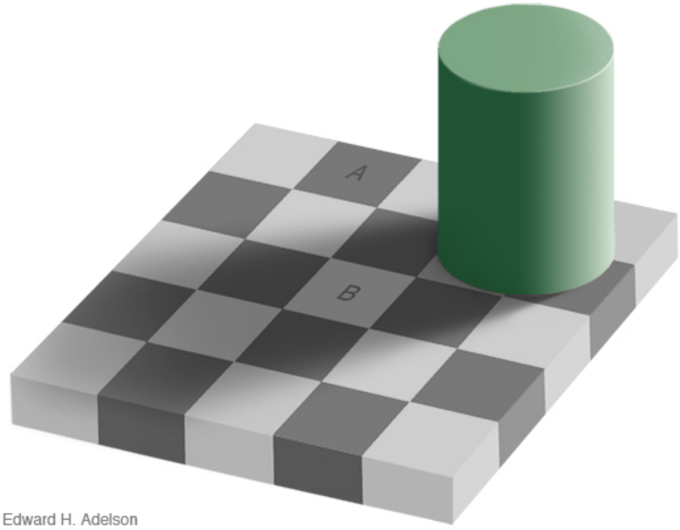
What information, then, can the qualities of such sensations give us about the characteristics of the external causes and influences which produce them? Only this: our sensations are **signs, not images***, of such characteristics.*

Illusions

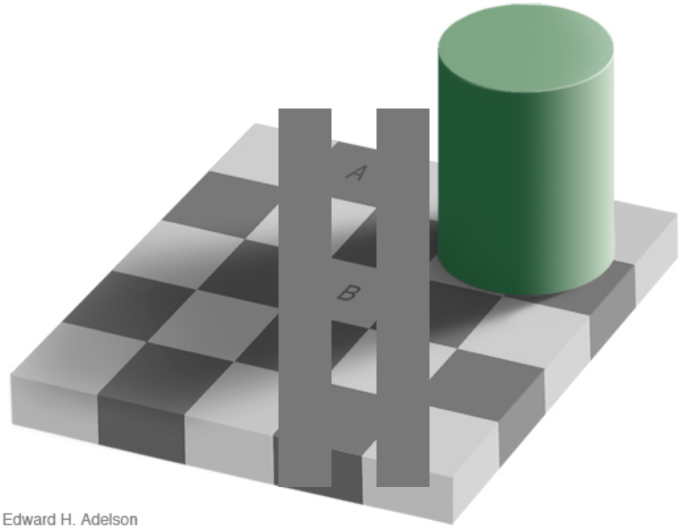


Gregory 1968

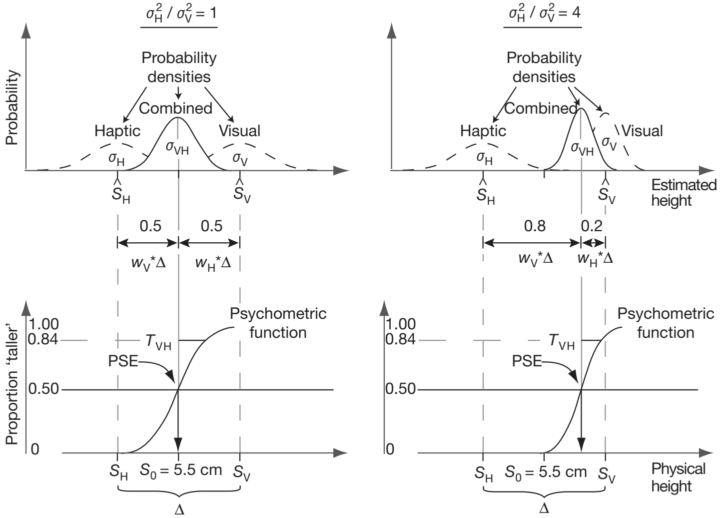
Illusions



Illusions

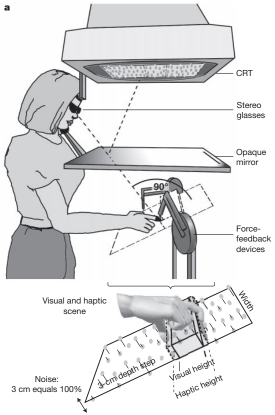


Cue combination



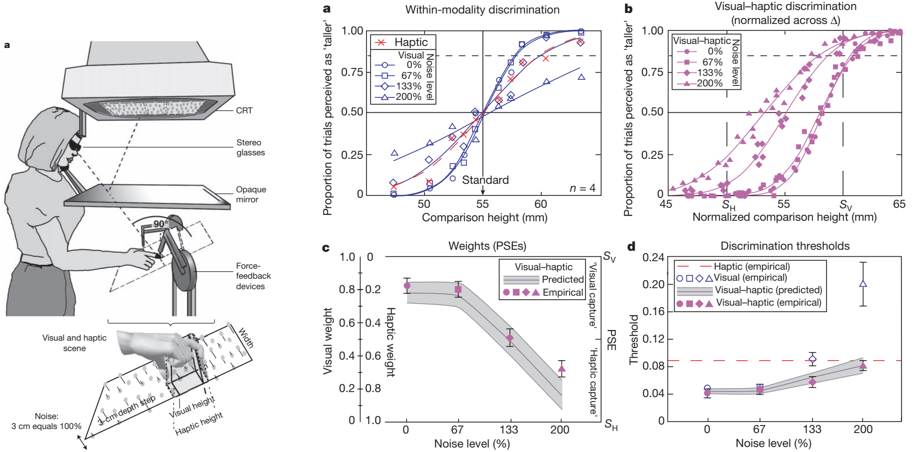
Ernst & Banks 2002

Cue combination



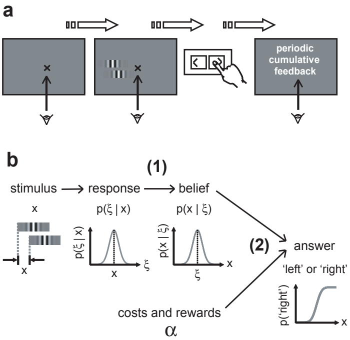
Ernst & Banks 2002

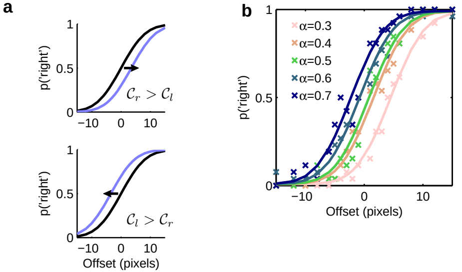
Cue combination



Ernst & Banks 2002

Bayesian Decisions

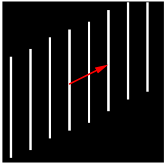




Sahani & Whiteley 2008

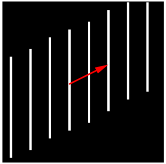
Motion Uncertainty

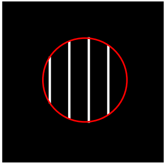
Given only local information about a moving edge, its velocity cannot be estimated. This is known as the aperture problem.



Motion Uncertainty

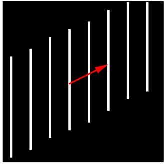
Given only local information about a moving edge, its velocity cannot be estimated. This is known as the aperture problem.

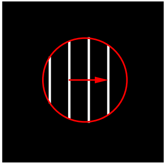




Motion Uncertainty

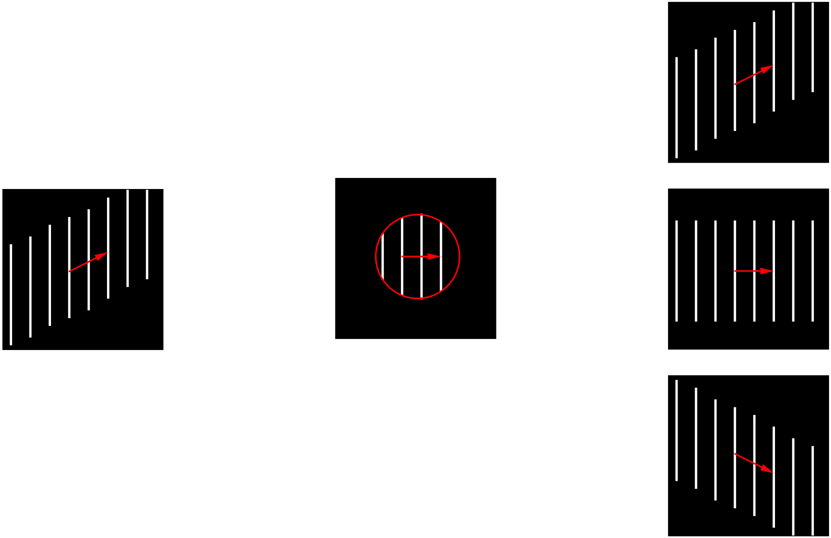
Given only local information about a moving edge, its velocity cannot be estimated. This is known as the aperture problem.





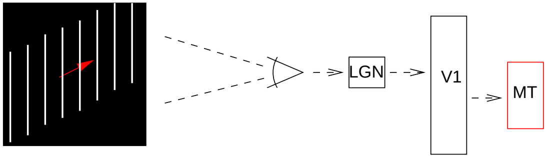
Motion Uncertainty

Given only local information about a moving edge, its velocity cannot be estimated. This is known as the aperture problem.



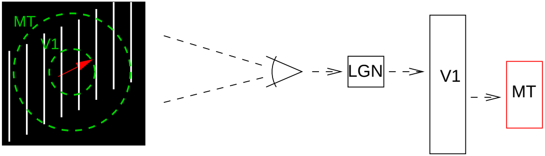
The Aperture Problem in V1 and MT

The aperture problem is relevant to the visual system because motion sensitive cells early in the visual pathway have small receptive fields. (Pack and Born 2001)



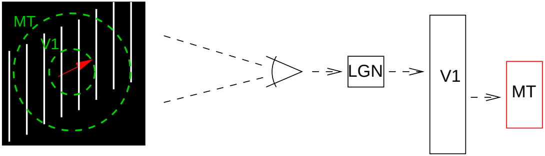
The Aperture Problem in V1 and MT

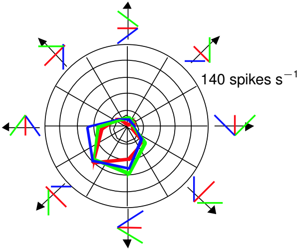
The aperture problem is relevant to the visual system because motion sensitive cells early in the visual pathway have small receptive fields. (Pack and Born 2001)



The Aperture Problem in V1 and MT

The aperture problem is relevant to the visual system because motion sensitive cells early in the visual pathway have small receptive fields. (Pack and Born 2001)

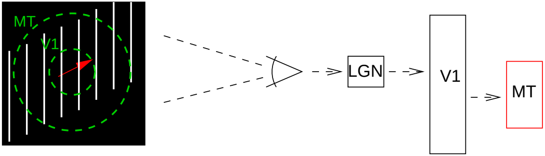


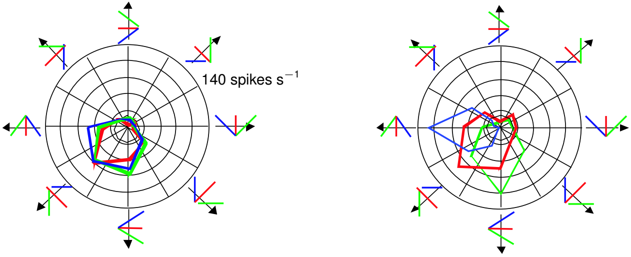


Late

The Aperture Problem in V1 and MT

The aperture problem is relevant to the visual system because motion sensitive cells early in the visual pathway have small receptive fields. (Pack and Born 2001)





Late

Early

Solving the Aperture Problem

The visual system appears to resolve the aperture problem. How does it do it?

- Local (aperture constrained) measurements must be combined with object form cues to estimate global object motion.

Solving the Aperture Problem

The visual system appears to resolve the aperture problem. How does it do it?

- Local (aperture constrained) measurements must be combined with object form cues to estimate global object motion.

- Two candidate algorithms
  - vector average
  - intersection of constraints

  but observers seem to switch from one to the other (or use intermediates) as other stimulus features change.

Solving the Aperture Problem

The visual system appears to resolve the aperture problem. How does it do it?

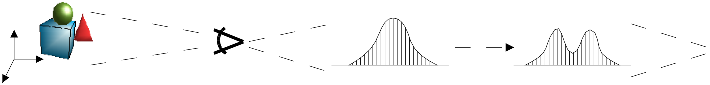
- Local (aperture constrained) measurements must be combined with object form cues to estimate global object motion.

- Two candidate algorithms
  - vector average
  - intersection of constraints

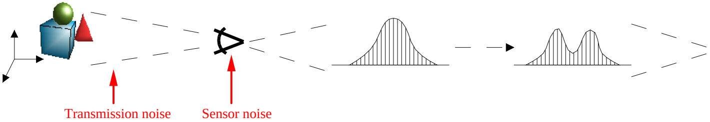
  but observers seem to switch from one to the other (or use intermediates) as other stimulus features change.

- Weiss and Adleson showed that the psychophysical evidence could be well modelled if observers were assumed to retain uncertainty about local estimates, and combine them, along with an *a priori* expectation, in a probabilistically appropriate fashion.

Uncertainty in Perception

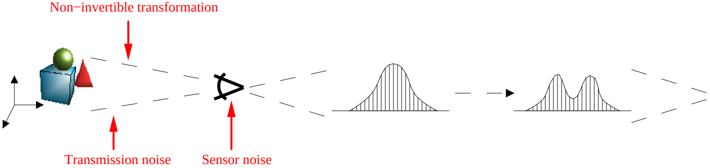


Uncertainty in Perception



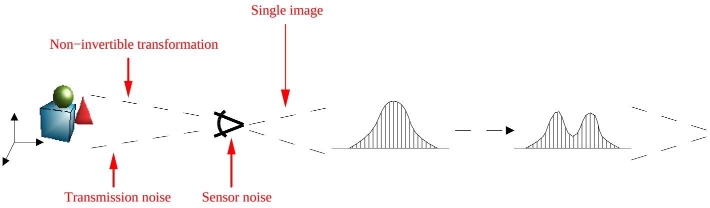
- Noise is added in transmission and at the sensor.

Uncertainty in Perception



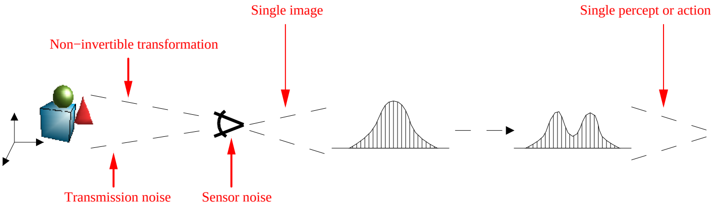
- Noise is added in transmission and at the sensor.
- The projection to the relatively low-dimensional sensor is usually non-invertible.

Uncertainty in Perception



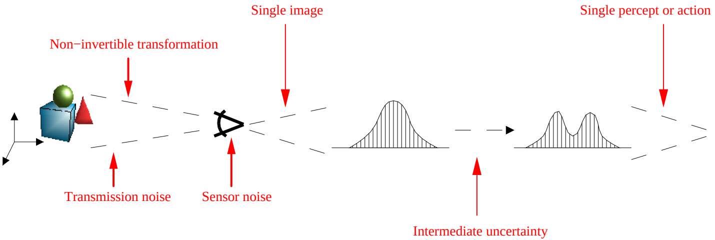
- Noise is added in transmission and at the sensor.
- The projection to the relatively low-dimensional sensor is usually non-invertible.
- The sensor experiences only a single image.

Uncertainty in Perception



- Noise is added in transmission and at the sensor.
- The projection to the relatively low-dimensional sensor is usually non-invertible.
- The sensor experiences only a single image.
- In general, the eventual percept or action is also unitary.

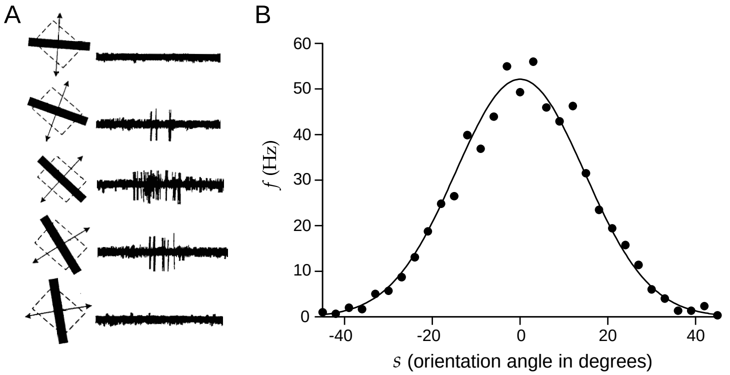
Uncertainty in Perception



- Noise is added in transmission and at the sensor.
- The projection to the relatively low-dimensional sensor is usually non-invertible.
- The sensor experiences only a single image.
- In general, the eventual percept or action is also unitary.
- Intermediate stages of computation require representation of distributions over various inferred "features".

Information Representation

Individual neurons are broadly tuned and noisy. Information appears to be conveyed by neuronal populations.



Population Codes

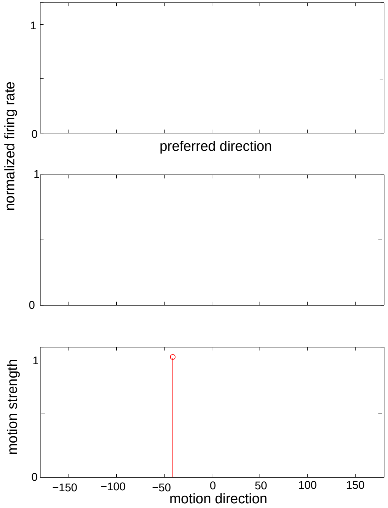
Information about single quantities can quite easily be recovered from a population code, even with noisy outputs.

Population Codes

Information about single quantities can quite easily be recovered from a population code, even with noisy outputs.

- Encoding:

  input = x

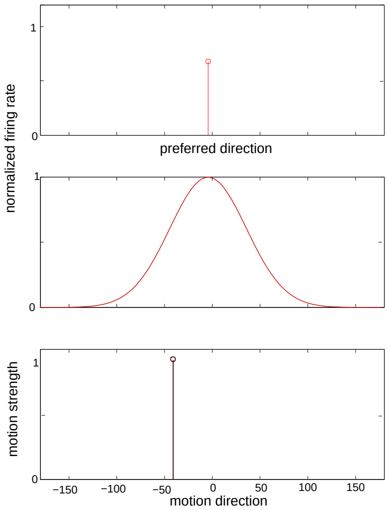


Population Codes

Information about single quantities can quite easily be recovered from a population code, even with noisy outputs.

- Encoding:

$$r_i(x) = f_i(x)$$

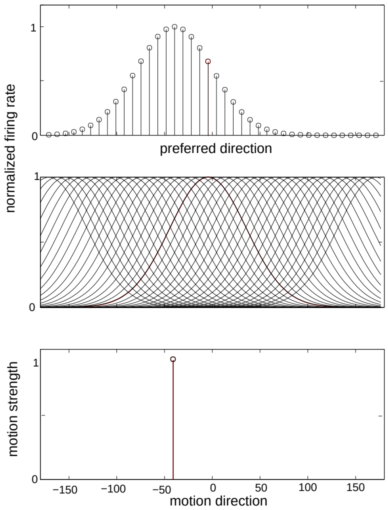


Population Codes

Information about single quantities can quite easily be recovered from a population code, even with noisy outputs.

- Encoding:

$$r_i(x) = f_i(x)$$

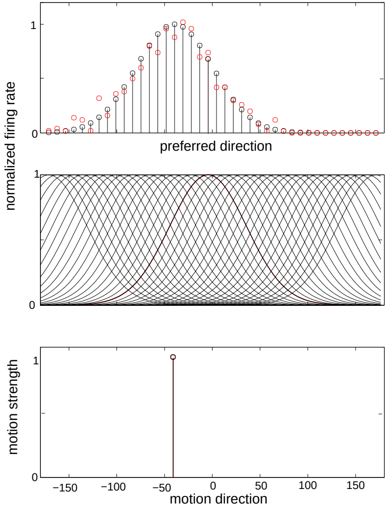


Population Codes

Information about single quantities can quite easily be recovered from a population code, even with noisy outputs.

- Encoding:

$$r_i(x) = f_i(x)$$
$$n_i \sim Poisson(r_i)$$



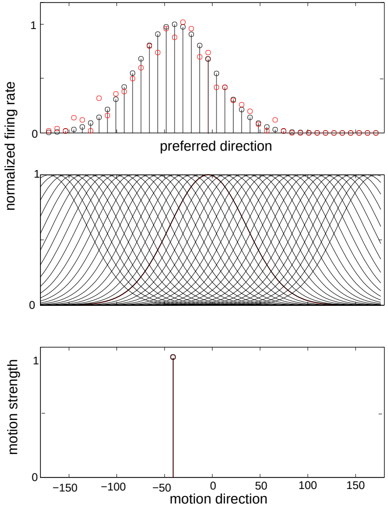
Population Codes

Information about single quantities can quite easily be recovered from a population code, even with noisy outputs.

- Encoding:

$$r_i(x) = f_i(x)$$
$$n_i \sim Poisson(r_i)$$

- Decoding:
  - Linear (Population Vector)



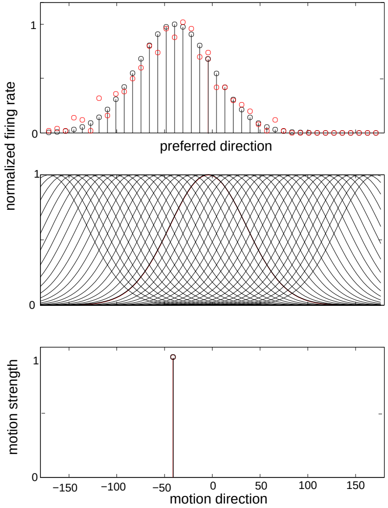
Population Codes

Information about single quantities can quite easily be recovered from a population code, even with noisy outputs.

- Encoding:

$$r_i(x) = f_i(x)$$
$$n_i \sim Poisson(r_i)$$

- Decoding:
  - Linear (Population Vector)
  - Maximum Likelihood



Linear Encoding of Functions

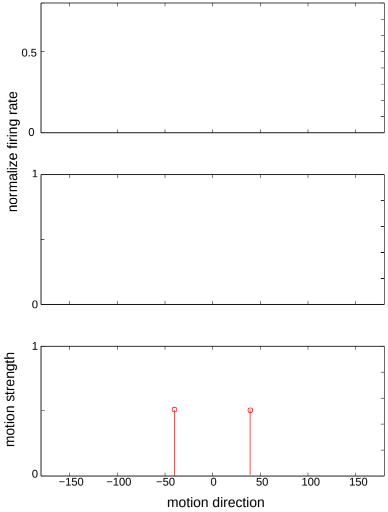
We can also encode a function over the stimulus dimension (a feature map) $m(x)$.

Linear Encoding of Functions

We can also encode a function over the stimulus dimension (a feature map) $m(x)$.

- Encoding:

$$\text{input} = \qquad m(x)$$

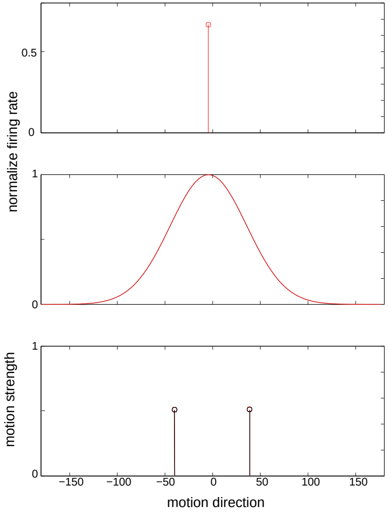


Linear Encoding of Functions

We can also encode a function over the stimulus dimension (a feature map) $m(x)$.

- Encoding:

$$r_i[m(x)] = \int dx \; f_i(x) m(x)$$

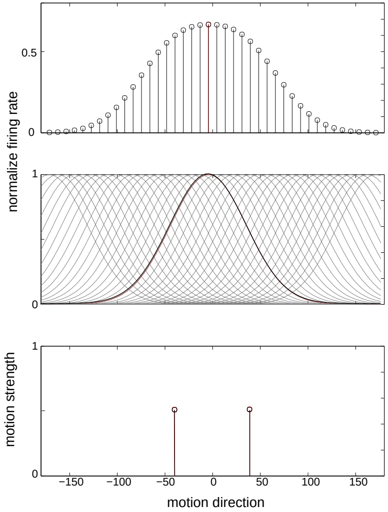


Linear Encoding of Functions

We can also encode a function over the stimulus dimension (a feature map) $m(x)$.

- Encoding:

$$r_i[m(x)] = \int dx\; f_i(x)m(x)$$



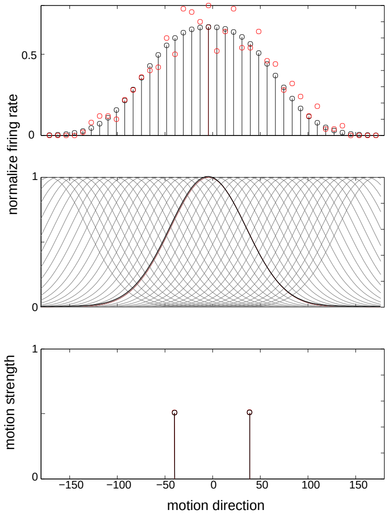
Linear Encoding of Functions

We can also encode a function over the stimulus dimension (a feature map) $m(x)$.

- Encoding:

$$r_i[m(x)] = \int dx \; f_i(x) m(x)$$

$$n_i \sim Poisson(r_i)$$



Linear Encoding of Functions

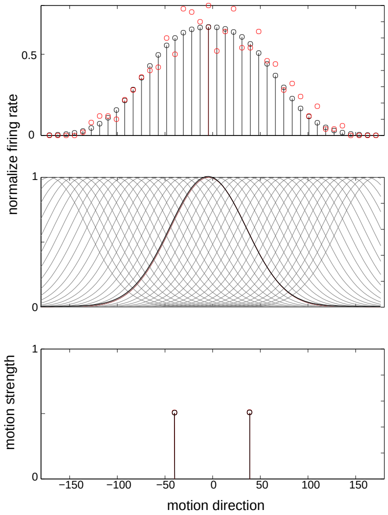
We can also encode a function over the stimulus dimension (a feature map) $m(x)$.

- Encoding:

$$r_i[m(x)] = \int dx \; f_i(x) m(x)$$

$$n_i \sim Poisson(r_i)$$

- Decoding:
  - Vector average returns only one value of x.



Linear Encoding of Functions

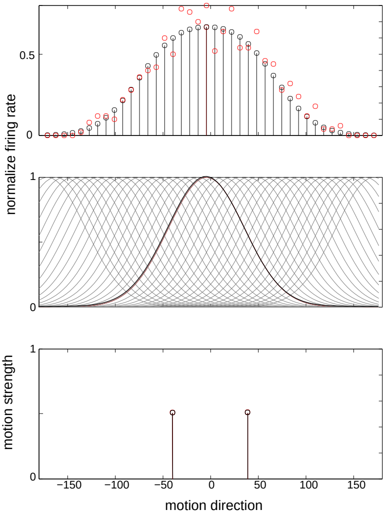
We can also encode a function over the stimulus dimension (a feature map) $m(x)$.

- Encoding:

$$r_i[m(x)] = \int dx\ f_i(x) m(x)$$

$$n_i \sim Poisson(r_i)$$

- Decoding:
  - Vector average returns only one value of x.
  - Linear basis functions (Anderson) do not exploit the full representational power.



Linear Encoding of Functions

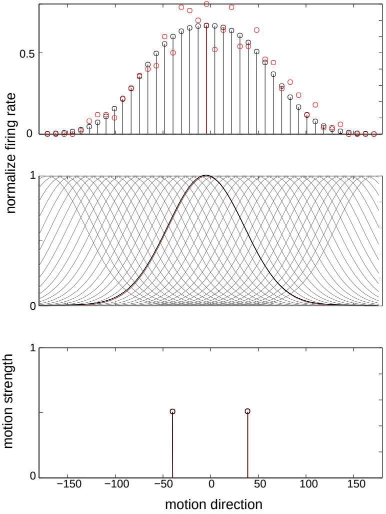
We can also encode a function over the stimulus dimension (a feature map) $m(x)$.

- Encoding:

$$r_i[m(x)] = \int dx\ f_i(x) m(x)$$

$$n_i \sim Poisson(r_i)$$

- Decoding:
  - Vector average returns only one value of x.
  - Linear basis functions (Anderson) do not exploit the full representational power.
  - Maximum likelihood (Zemel *et al.*) is powerful, but expensive.



Representing uncertainty

- Deterministic representations
  - Linear decoding
  - Linear encoding (DPC)
  - Log-linear decoding (natural parameters)
  - 'Probabilistic encoding' / Inferential decoding (PPC)
- Stochastic (sample-based) representations

Linear decoding

$$p(x; \mathbf{r}) \propto \Big[\sum_a \phi_a(x) r_a \Big]_+$$

- Discussed by Anderson (90s); recent work by EliasSmith and others.
- Computations linear in probability / density become easy.
- Encoding may be difficult.
- Basis functions ϕ_a set a bound on possible precision.
- Noise enters decoder directly – suppressed if uncorrelated.

Linear encoding

$$r_a = \left[\int dx \, \phi_a(x) p(x) \right]_+ = [\langle \phi_a(x) \rangle]_+$$

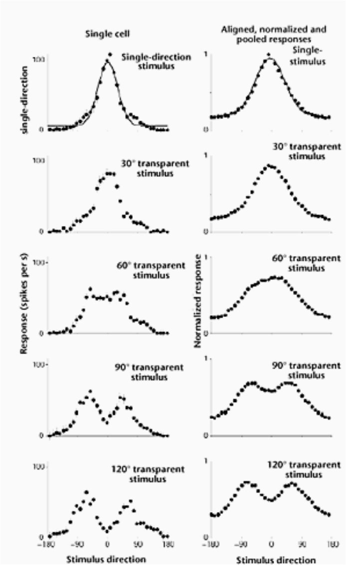
- "Distributional Population Code" (DPC) – Pouget, Zemel, Dayan.
- Encoding easy to learn (delta rule)
- Decoding (i.e. identifying natural parameters) may be challenging – MaxEnt or EM-like algorithm if rates are noisy.
- Computation must be learnt.

Transparency or Uncertainty?

Density functions can represent either simulataneous presence (transparency) or alternative presence (uncertainty).

Transparency or Uncertainty?

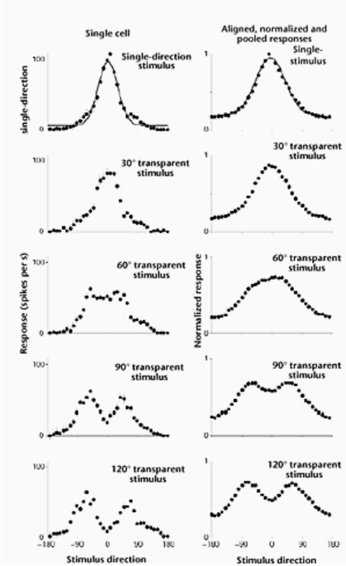
Density functions can represent either simulataneous presence (transparency) or alternative presence (uncertainty).



- Function coding as described seems to model codes for transparent motion well.

Transparency or Uncertainty?

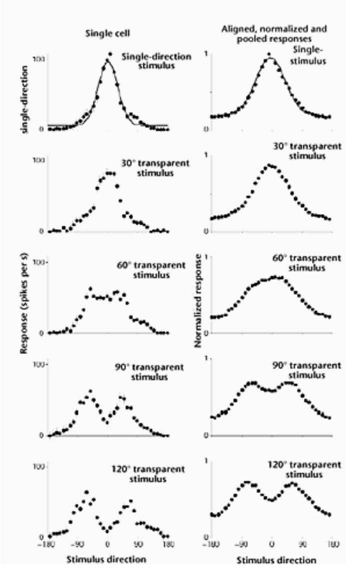
Density functions can represent either simulataneous presence (transparency) or alternative presence (uncertainty).



- Function coding as described seems to model codes for transparent motion well.
- Cannot represent uncertainty about stimulus presence.

Transparency or Uncertainty?

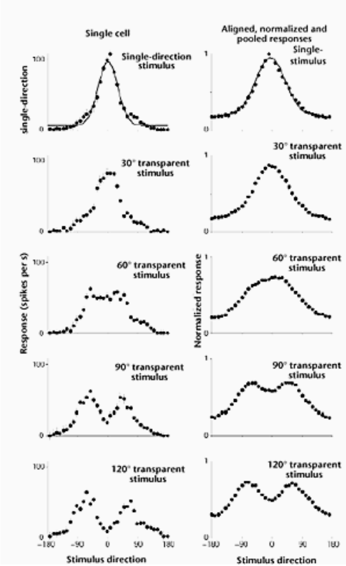
Density functions can represent either simulataneous presence (transparency) or alternative presence (uncertainty).



- Function coding as described seems to model codes for transparent motion well.
- Cannot represent uncertainty about stimulus presence.
- Cannot represent uncertainty about a multiple stimulus.

Transparency or Uncertainty?

Density functions can represent either simulataneous presence (transparency) or alternative presence (uncertainty).



- Function coding as described seems to model codes for transparent motion well.
- Cannot represent uncertainty about stimulus presence.
- Cannot represent uncertainty about a multiple stimulus.
- So then what about uncertainty?

Uncertainty over Feature Maps

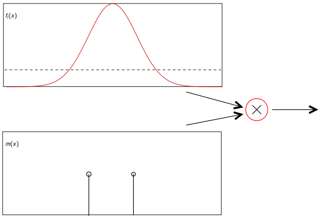
The solution is to encode the uncertainty about the entire feature *map* $m(x)$.
(This uncertainty is described by a probability functional, $p[m]$.)

$$r_i[\quad] = \qquad ?$$

Uncertainty over Feature Maps

The solution is to encode the uncertainty about the entire feature *map* $m(x)$.
(This uncertainty is described by a probability functional, $p[m]$.)

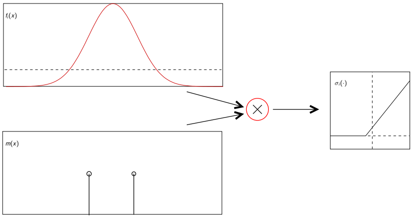
$$r_i[m] = \int dx \; f_i(x) m(x)$$



Uncertainty over Feature Maps

The solution is to encode the uncertainty about the entire feature *map* $m(x)$.
(This uncertainty is described by a probability functional, $p[m]$.)

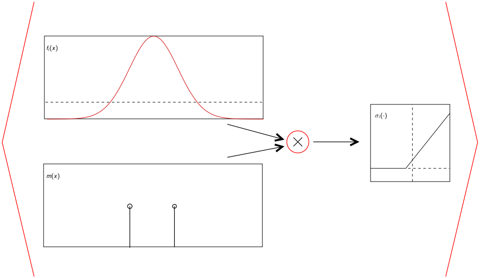
$$r_i[m] = \quad \sigma_i\left(\int dx \; f_i(x) m(x) \right)$$



Uncertainty over Feature Maps

The solution is to encode the uncertainty about the entire feature *map* $m(x)$.
(This uncertainty is described by a probability functional, $p[m]$.)

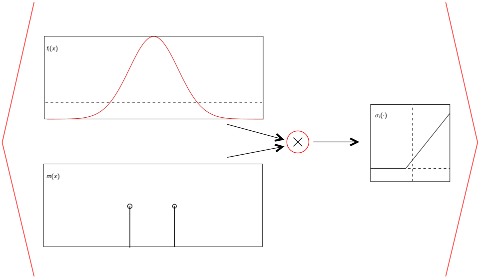
$$r_i[\,p\,] = \left\langle \sigma_i\left(\int dx\ f_i(x) m(x) \right) \right\rangle_{p[m]}$$



Uncertainty over Feature Maps

The solution is to encode the uncertainty about the entire feature *map* $m(x)$.
(This uncertainty is described by a probability functional, $p[m]$.)

$$r_i[\,p\,] = \Big\langle \underbrace{\sigma_i\Big(\int dx\ f_i(x)m(x)\Big)}_{\psi_i[m]} \Big\rangle_{p[m]}$$



Why the Expected Firing Rate?

$$r_i[\,p\,] = \left\langle \sigma_i\left(\int dx\ f_i(x)m(x)\right)\right\rangle_{p[m]}$$

Why the Expected Firing Rate?

$$r_i[p] = \left\langle \sigma_i\left(\int dx\ f_i(x) m(x)\right) \right\rangle_{p[m]}$$

- Sufficient to represent uncertainty (will be shown later).

Why the Expected Firing Rate?

$$r_i[p] = \left\langle \sigma_i\left(\int dx\; f_i(x)m(x)\right)\right\rangle_{p[m]}$$

- Sufficient to represent uncertainty (will be shown later).
- Easy to learn from example feature maps drawn from $p[m]$.

Why the Expected Firing Rate?

$$r_i[\,p\,] = \left\langle \sigma_i\left(\int dx\ f_i(x)m(x)\right) \right\rangle_{p[m]}$$

- Sufficient to represent uncertainty (will be shown later).
- Easy to learn from example feature maps drawn from $p[m]$.
- Matches the intuitive notion that the firing rate of an “indicator” neuron signals confidence.

Why the Expected Firing Rate?

$$r_i[\,p\,] = \left\langle \sigma_i\left(\int dx\; f_i(x)m(x)\right) \right\rangle_{p[m]}$$

- Sufficient to represent uncertainty (will be shown later).
- Easy to learn from example feature maps drawn from $p[m]$.
- Matches the intuitive notion that the firing rate of an “indicator” neuron signals confidence.
- Reduces to conventional single feature and function encoding schemes in the appropriate limits.

Why a Nonlinear Transfer Function?

$$r_i[\,p\,] = \left\langle \sigma_i\left(\int dx \ f_i(x) m(x) \right) \right\rangle_{p[m]}$$

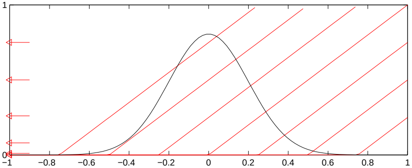
- Multiple different non-linearities exploit the overcomplete representation to form a (cumulative) “histogram” of the distribution.

Why a Nonlinear Transfer Function?

$$r_i[\,p\,] = \left\langle \sigma_i\left(\int dx\ f_i(x) m(x) \right) \right\rangle_{p[m]}$$

- Multiple different non-linearities exploit the overcomplete representation to form a (cumulative) "histogram" of the distribution.

$$\begin{aligned} r_i[p] &= \langle \sigma_i(m) \rangle_{p(m)} \\ &= \int dm\ p(m) \sigma_i(m) \end{aligned}$$

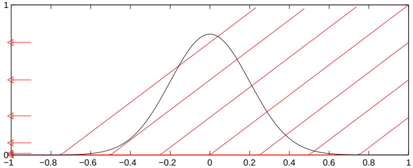


Why a Nonlinear Transfer Function?

$$r_i[p] = \left\langle \sigma_i\left(\int dx\ f_i(x) m(x) \right) \right\rangle_{p[m]}$$

- Multiple different non-linearities exploit the overcomplete representation to form a (cumulative) "histogram" of the distribution.

$$r_i[p] = \langle \sigma_i(m) \rangle_{p(m)}$$
$$= \int dm\ p(m) \sigma_i(m)$$



- A linear transfer function would only encode the mean feature map:

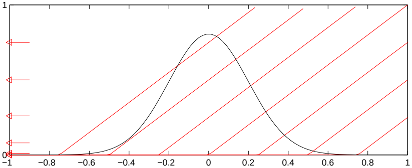
$$\left\langle \int dx\ f_i(x) m(x) \right\rangle_{p[m]} = \int dx\ f_i(x) \langle m(x) \rangle_{p[m]}$$

Why a Nonlinear Transfer Function?

$$r_i[p] = \left\langle \sigma_i\left(\int dx\ f_i(x) m(x) \right) \right\rangle_{p[m]}$$

- Multiple different non-linearities exploit the overcomplete representation to form a (cumulative) "histogram" of the distribution.

$$r_i[p] = \langle \sigma_i(m) \rangle_{p(m)} = \int dm\ p(m) \sigma_i(m)$$



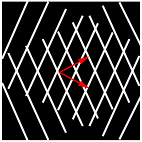
- A linear transfer function would only encode the mean feature map:

$$\left\langle \int dx\ f_i(x) m(x) \right\rangle_{p[m]} = \int dx\ f_i(x) \langle m(x) \rangle_{p[m]}$$

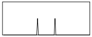
- For additional theoretical reasons, it is likely that some variation in transfer function between neurons is important.

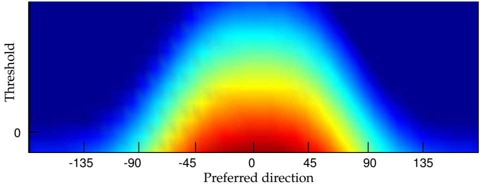
Representing Transparency and Uncertainty

Transparency:

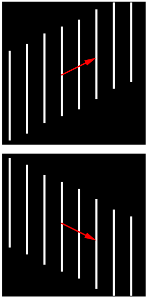


$$p[m_{12}] = 1$$

$m_{12} =$ 



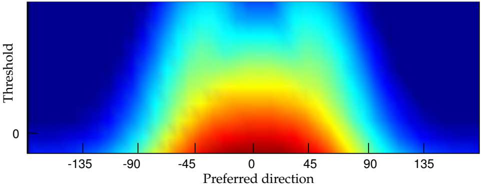
Uncertainty:



$$p[m_1] = p[m_2] = \frac{1}{2}$$

$m_1 =$ 

$m_2 =$ 



Decoding

Transparent and uncertain motion lead to visually different codes, but can they be recovered from the population?

- We decode by finding an estimated distribution $q[m]$ that best explains the observed firing rates.

NB: decoding (unlike computation) is not an operation of intrinsic biological interest; it merely demonstrates that the encoded information can be recovered in principle.

Decoding

Transparent and uncertain motion lead to visually different codes, but can they be recovered from the population?

- We decode by finding an estimated distribution $q[m]$ that best explains the observed firing rates.
- In the absence of noise, each observed rate places a single constraint on $q[m]$.

NB: decoding (unlike computation) is not an operation of intrinsic biological interest; it merely demonstrates that the encoded information can be recovered in principle.

Decoding

Transparent and uncertain motion lead to visually different codes, but can they be recovered from the population?

- We decode by finding an estimated distribution $q[m]$ that best explains the observed firing rates.
- In the absence of noise, each observed rate places a single constraint on $q[m]$.
  - Finite set of constraints insufficient to uniquely identify $q[m]$.

NB: decoding (unlike computation) is not an operation of intrinsic biological interest; it merely demonstrates that the encoded information can be recovered in principle.

Decoding

Transparent and uncertain motion lead to visually different codes, but can they be recovered from the population?

- We decode by finding an estimated distribution $q[m]$ that best explains the observed firing rates.
- In the absence of noise, each observed rate places a single constraint on $q[m]$.
  - Finite set of constraints insufficient to uniquely identify $q[m]$.
  - Choose the most uncertain (maximum entropy) $q[m]$ consistent with the constraints.

NB: decoding (unlike computation) is not an operation of intrinsic biological interest; it merely demonstrates that the encoded information can be recovered in principle.

Decoding

Transparent and uncertain motion lead to visually different codes, but can they be recovered from the population?

- We decode by finding an estimated distribution $q[m]$ that best explains the observed firing rates.
- In the absence of noise, each observed rate places a single constraint on $q[m]$.
  - Finite set of constraints insufficient to uniquely identify $q[m]$.
  - Choose the most uncertain (maximum entropy) $q[m]$ consistent with the constraints.
  - Well known problem solved by “generalized iterative scaling”.

NB: decoding (unlike computation) is not an operation of intrinsic biological interest; it merely demonstrates that the encoded information can be recovered in principle.

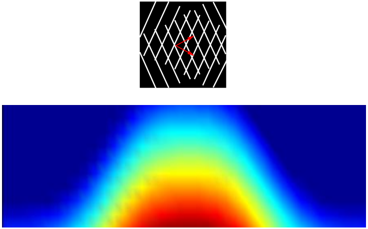
Decoding

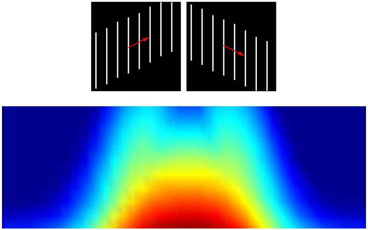
Transparent and uncertain motion lead to visually different codes, but can they be recovered from the population?

- We decode by finding an estimated distribution $q[m]$ that best explains the observed firing rates.
- In the absence of noise, each observed rate places a single constraint on $q[m]$.
  - Finite set of constraints insufficient to uniquely identify $q[m]$.
  - Choose the most uncertain (maximum entropy) $q[m]$ consistent with the constraints.
  - Well known problem solved by “generalized iterative scaling”.
- Realistically, only a noisy estimate of the rate will be available.
  - Constraints cannot be satisfied exactly.
  - Find $q[m]$ for which the observed spike counts are most likely.
  - Decoding in this case tests the robustness of the code to noise.

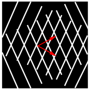
NB: decoding (unlike computation) is not an operation of intrinsic biological interest; it merely demonstrates that the encoded information can be recovered in principle.

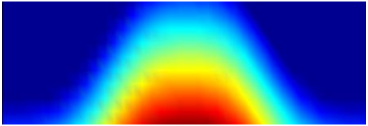
Transparency vs. Uncertainty

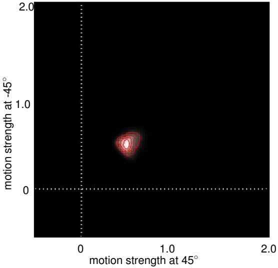


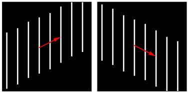


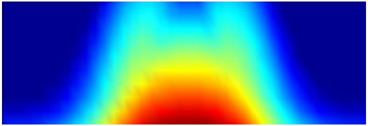
Transparency vs. Uncertainty



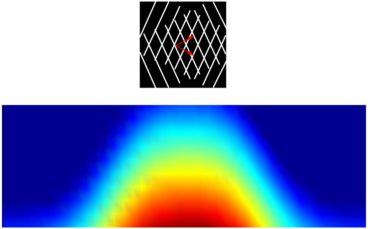


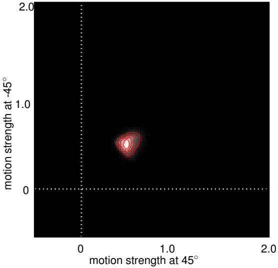


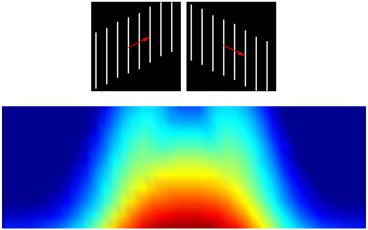


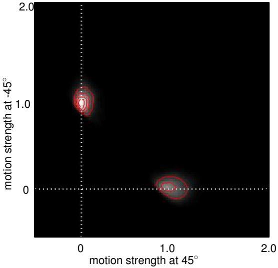


Transparency vs. Uncertainty

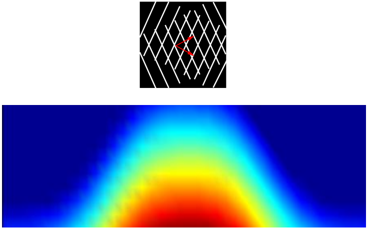


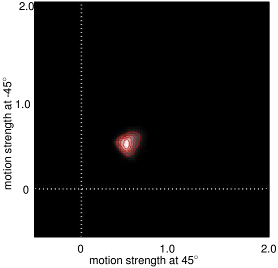


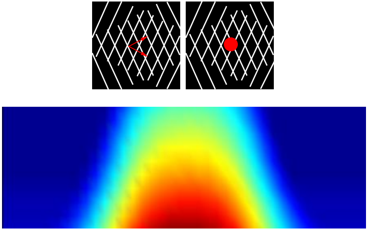




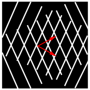
Uncertainty about Stimulus Presence

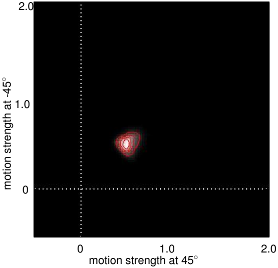


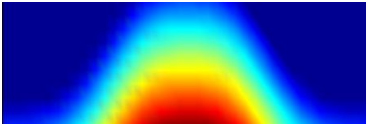


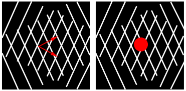


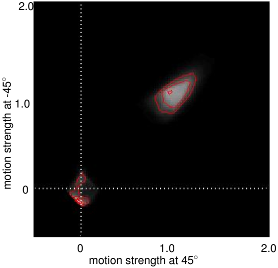
Uncertainty about Stimulus Presence

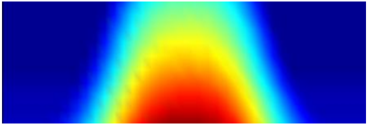












Why does the DDPC work?

Why does the DDPC work?

- The maximum entropy distribution associated with expected value constraints is a member of the exponential family:

$$q[m] = \frac{1}{Z} e^{\sum_i \theta_i \psi_i[m]}$$

Why does the DDPC work?

- The maximum entropy distribution associated with expected value constraints is a member of the exponential family:

$$q[m] = \frac{1}{Z} e^{\sum_i \theta_i \psi_i[m]}$$

- Exponential family distributions can (in general) be parametrised equally well by:
  - the *natural parameters* θ_i
  - or the mean parameters $\mathbb{E}_q[\psi_i]$

Why does the DDPC work?

- The maximum entropy distribution associated with expected value constraints is a member of the exponential family:

$$q[m] = \frac{1}{Z} e^{\sum_i \theta_i \psi_i[m]}$$

- Exponential family distributions can (in general) be parametrised equally well by:
  - the *natural parameters* θ_i
  - or the mean parameters $\mathbb{E}_q[\psi_i] \leftarrow r_i$!

Why does the DDPC work?

- The maximum entropy distribution associated with expected value constraints is a member of the exponential family:

$$q[m] = \frac{1}{Z} e^{\sum_i \theta_i \psi_i[m]}$$

- Exponential family distributions can (in general) be parametrised equally well by:
  - the *natural parameters* θ_i
  - or the mean parameters $\mathbb{E}_q[\psi_i] \leftarrow r_i$!

- *Decoding* means obtaining the natural parameters from the mean parameters [the natural paremeters are needed to evaluate the (unnormalised) density].

Why does the DDPC work?

- The maximum entropy distribution associated with expected value constraints is a member of the exponential family:

$$q[m] = \frac{1}{Z} e^{\sum_i \theta_i \psi_i[m]}$$

- Exponential family distributions can (in general) be parametrised equally well by:
  - the *natural parameters* θ_i
  - or the mean parameters $\mathbb{E}_q[\psi_i] \leftarrow r_i$!

- *Decoding* means obtaining the natural parameters from the mean parameters [the natural paremeters are needed to evaluate the (unnormalised) density].

  This is a convex problem with a unique solution!

Why does the DDPC work?

- The maximum entropy distribution associated with expected value constraints is a member of the exponential family:

$$q[m] = \frac{1}{Z} e^{\sum_i \theta_i \psi_i[m]}$$

- Exponential family distributions can (in general) be parametrised equally well by:
  - the *natural parameters* θ_i
  - or the mean parameters $\mathbb{E}_q[\psi_i] \leftarrow r_i$!

- *Decoding* means obtaining the natural parameters from the mean parameters [the natural paremeters are needed to evaluate the (unnormalised) density].

  This is a convex problem with a unique solution!

  Typically still hard.

Why does the DDPC work?

- The maximum entropy distribution associated with expected value constraints is a member of the exponential family:

$$q[m] = \frac{1}{Z} e^{\sum_i \theta_i \psi_i[m]}$$

- Exponential family distributions can (in general) be parametrised equally well by:
  - the *natural parameters* θ_i
  - or the mean parameters $\mathbb{E}_q[\psi_i] \leftarrow r_i$!

- *Decoding* means obtaining the natural parameters from the mean parameters [the natural paremeters are needed to evaluate the (unnormalised) density].

  This is a convex problem with a unique solution!

  Typically still hard.

- But: decoding (unlike computation) is not an operation of intrinsic biological interest.

Log-linear decoding

$$p(x; \mathbf{r}) \propto \exp\left(\sum_a \phi_a(x) r_a\right)$$

- Natural parameter encoding.
- Makes some computations (e.g. cue combination) very easy.
- Encoding may be difficult to learn.
- Uncorrelated noise in activities may average away.
- Basis functions set maximum log-precision.

PPC

$$p(x; \mathbf{r}) = p(x|\mathbf{r}) \propto p(\mathbf{r}|x)p(x)$$

- For 'Poisson-like' $p(\mathbf{r}|x)$ (linear sufficient stats) this gives log-linear decoding.
- Unclear what $p(\mathbf{r}|x)$ should be – often taken to be observed experimental noise, but this is incorrect.
- Confuses information content with encoding: does retinal activity "encode" everything about a scene?
- "Representation" depends on knowing both likelihood and prior $\Rightarrow$ will usually depend on knowledge of the external world.
- Fixing a likelihood and prior is equivalent to assuming a parametric encoding.

Computation with DDCs

- Many (even most) probabilistic computations depend on calculating expectations.
  - Conditional marginalisation (prediction, message passing):

$$p(x) = \int dy\, p(x|y)p(y) = \mathbb{E}_{p(y)}\left[p(x|y)\right]$$

  - Variational (EM) learning in latent variable models:

$$\theta^{\text{new}} = \operatorname{argmax} \mathbb{E}_{q(x^{\text{lat}})}\left[\log p(x^{\text{obs}}, x^{\text{lat}}|\theta)\right]$$

  - Action evaluation (Bayesian decision theory)

$$Q(a, b) = \mathbb{E}_{b(s)}\left[Q(a, s)\right]$$

Computation with DDCs

- Many (even most) probabilistic computations depend on calculating expectations.
  - Conditional marginalisation (prediction, message passing):

$$p(x) = \int dy\, p(x|y)p(y) = \mathbb{E}_{p(y)}\left[p(x|y)\right]$$

  - Variational (EM) learning in latent variable models:

$$\theta^{\text{new}} = \operatorname{argmax} \mathbb{E}_{q(x^{\text{lat}})}\left[\log p(x^{\text{obs}}, x^{\text{lat}}|\theta)\right]$$

  - Action evaluation (Bayesian decision theory)

$$Q(a, b) = \mathbb{E}_{b(s)}\left[Q(a, s)\right]$$

- If the $\psi_i[m]$ form an adequate basis for the required functions of m, then these expectations can be computed as linear combinations of r_i:

$$f[m] = \sum_i \alpha_i \psi_i[m]$$

$$\Rightarrow \mathbb{E}\left[f[m]\right] = \sum_i \alpha_i \mathbb{E}\left[\psi_i[m]\right] = \sum_i \alpha_i r_i$$

Computation with DDCs

- Many (even most) probabilistic computations depend on calculating expectations.
  - Conditional marginalisation (prediction, message passing):

$$p(x) = \int dy\, p(x|y)p(y) = \mathbb{E}_{p(y)}\left[p(x|y)\right]$$

  - Variational (EM) learning in latent variable models:

$$\theta^{\text{new}} = \operatorname{argmax} \mathbb{E}_{q(x^{\text{lat}})}\left[\log p(x^{\text{obs}}, x^{\text{lat}}|\theta)\right]$$

  - Action evaluation (Bayesian decision theory)

$$Q(a, b) = \mathbb{E}_{b(s)}\left[Q(a, s)\right]$$

- If the $\psi_i[m]$ form an adequate basis for the required functions of m, then these expectations can be computed as linear combinations of r_i:

$$f[m] = \sum_i \alpha_i \psi_i[m]$$

$$\Rightarrow \mathbb{E}\left[f[m]\right] = \sum_i \alpha_i \mathbb{E}\left[\psi_i[m]\right] = \sum_i \alpha_i r_i$$

- Cue or message combination may be complex.
- [Related to belief states, predictive state representations and RKHS mean embeddings; c.f. Kernel Belief Propagation (Song et al 2011)]

Computation with log-likelihood codes

- cue (message) combination $\Rightarrow$ addition.
- projection / marginalisation? [see work by Beck et al.]