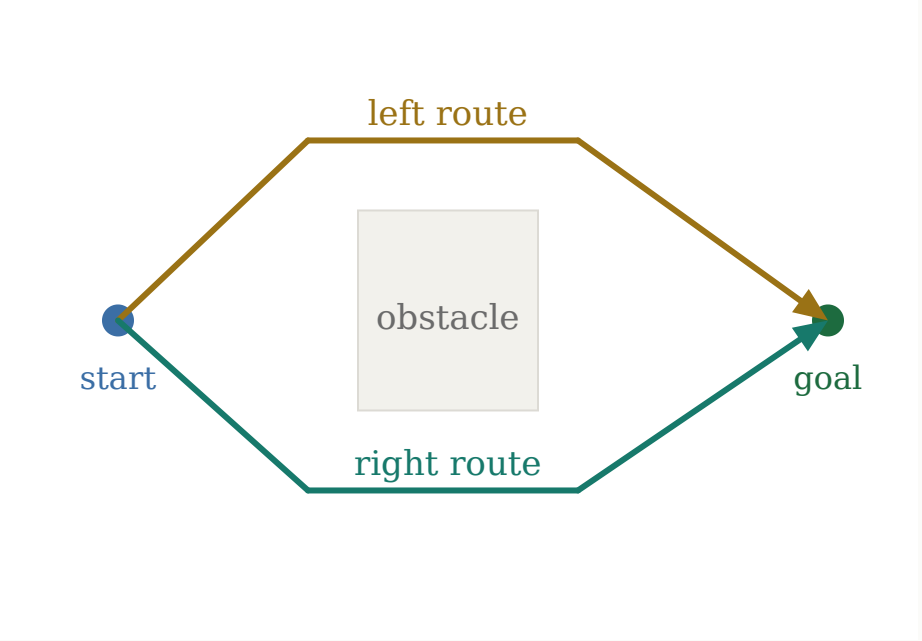
LECTURE 9 · NOVEMBER 11, 2026

Information Theory, Sampling and Density

How should a robot compare uncertain predictions?

Two safe routes can have different probabilities. A useful model must represent both, and we need evidence that it does.



Mathematics for Robotics · ROB-GY 6013
NYU Tandon · Jing Zhang · Fall 2026

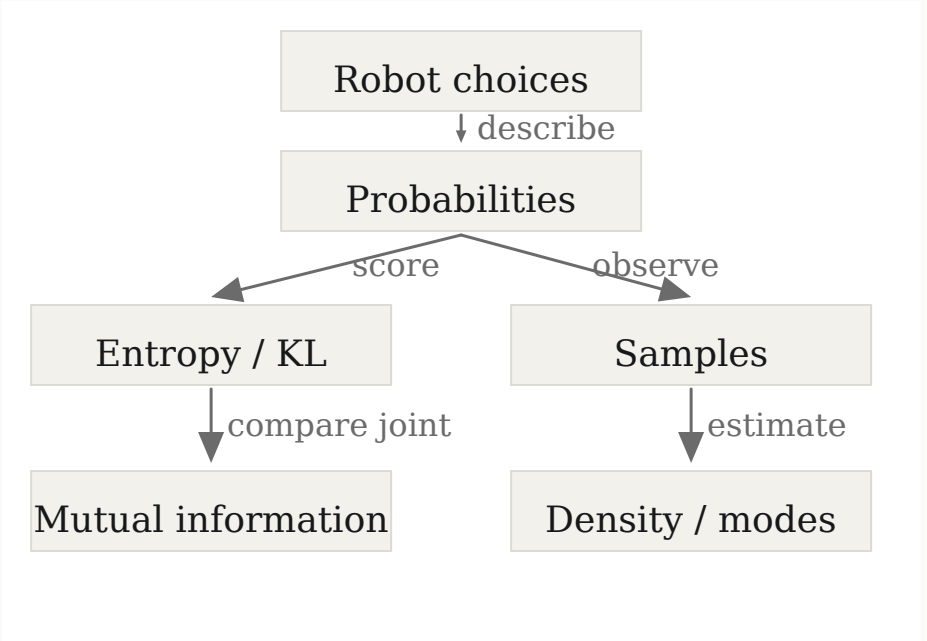
Goodfellow et al., Deep Learning (2016), §3.13

CONCEPT MAP

The reasoning path

One robot problem, several different questions

A distribution describes uncertainty. A score compares predictions. Samples estimate quantities. A density model reveals structure.



Goodfellow et al., Deep Learning (2016), §3.13

RECALL

Probability recall

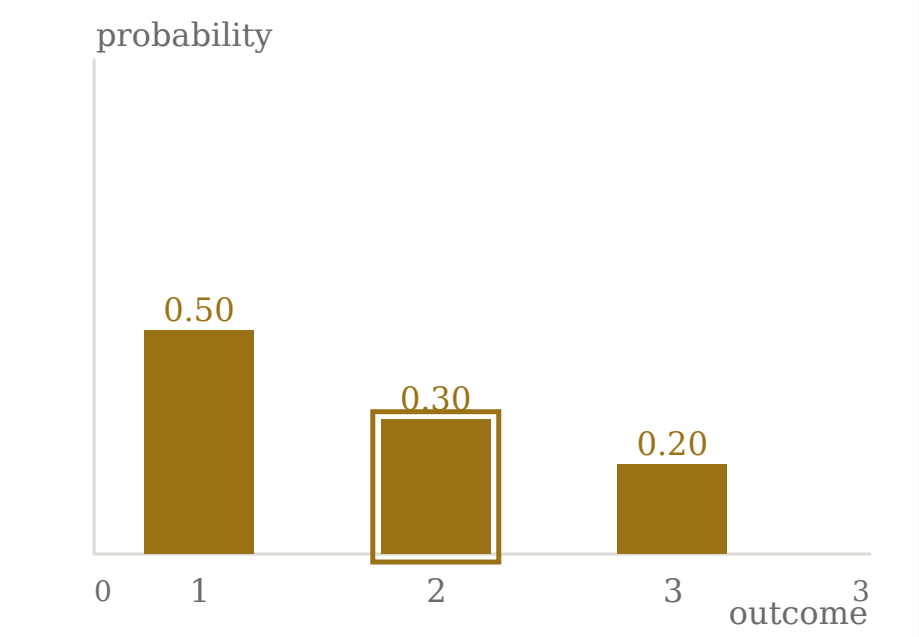
The distribution and its samples are different objects

Assume a finite outcome set. The probabilities are fixed; the observed outcomes vary.

$$\mathbb{E}_{p}[g(X)] = \sum_{x} p(x) g(x)$$

`X,x` random outcome and possible value

`p,g` probability mass function and measured quantity



Goodfellow et al., Deep Learning (2016), §3.8; original example

DEFINITION

Surprisal

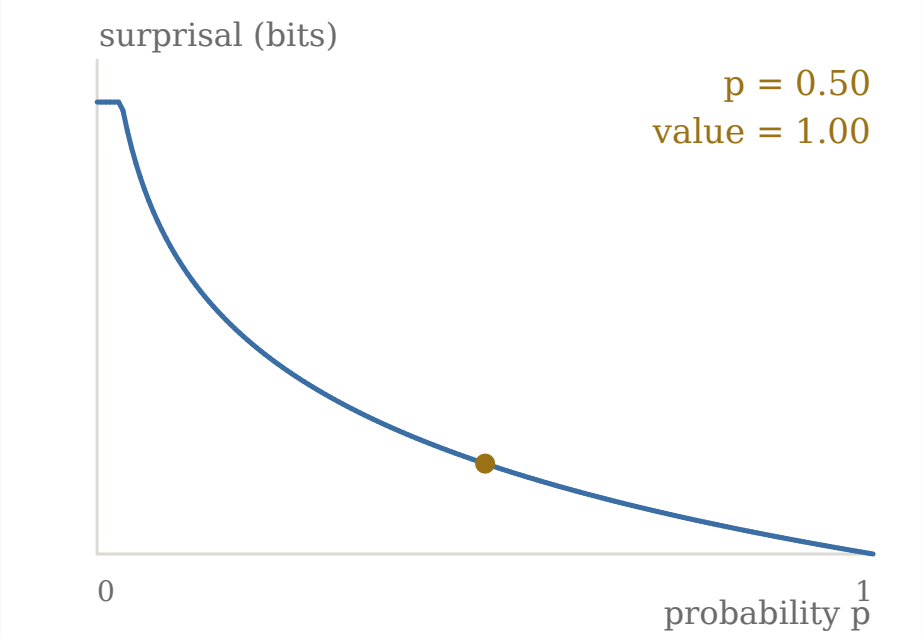
A confident mistake should cost more

A predicted probability of 0.01 makes an observed event more surprising than a prediction of 0.5.

$$I_q(x) = -\log_2 q(x)$$

`q(x)` predicted probability of the observed event

`I_q(x)` surprisal, measured in bits



Goodfellow et al., Deep Learning (2016), §3.13

DERIVATION

Surprisal

Independent evidence adds in log space

For independent events, their joint probability is the product.

$$-\log_2(q_1 q_2) = -\log_2 q_1 - \log_2 q_2$$

`q_1,q_2` probabilities of independent events

Goodfellow et al., Deep Learning (2016), §3.13

Surprisal

Independent evidence adds in log space

Factor the independent joint probability

q_1 and q_2 are positive probabilities of independent events.

1. $q(x_1, x_2) = q_1 q_2$ by independence.
2. $I_q(x_1, x_2) = -\log_2(q_1 q_2)$ by the definition of surprise.
3. $-\log_2(q_1 q_2) = -\log_2 q_1 - \log_2 q_2$ by the log-product rule.

Goodfellow et al., Deep Learning (2016), §3.13

DEFINITION

Entropy

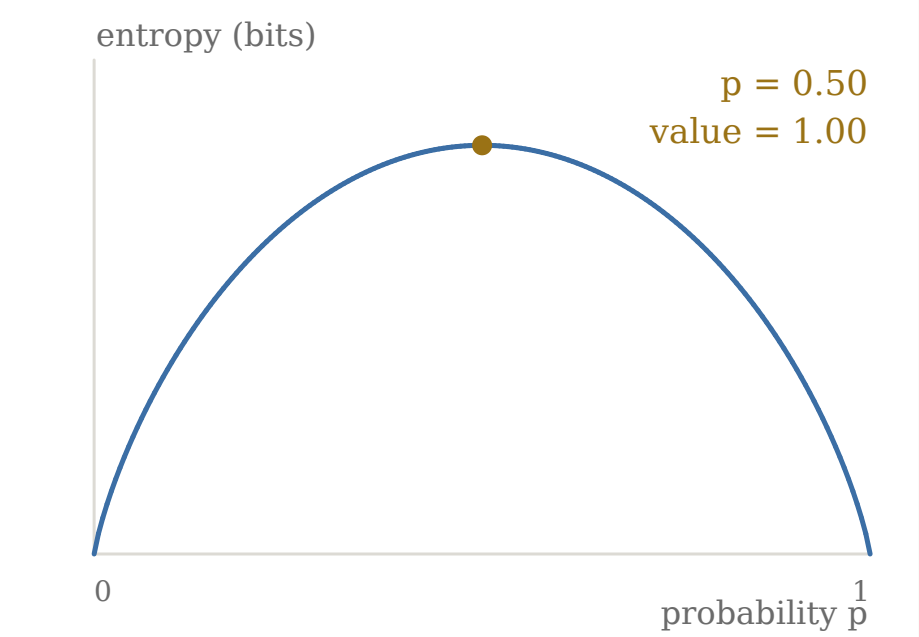
Average surprise under the true distribution

Use the same distribution to generate and score the outcome.

$$H(p) = -\sum_{x} p(x) \log_2 p(x)$$

`p(x)` probability of outcome x

`H(p)` entropy in bits; use 0 log 0 = 0



Goodfellow et al., Deep Learning (2016), §3.13

WORKED EXAMPLE

Entropy

A concrete prefix code makes the units visible

COURSE-CREATED WORKED EXAMPLE

Four symbols have probabilities 1/2, 1/4, 1/8, 1/8. Use codewords 0, 10, 110, 111.

$$L = \tfrac{1}{2}(1) + \tfrac{1}{4}(2) + \tfrac{1}{8}(3) + \tfrac{1}{8}(3)$$

L expected codeword length in bits

symbol	probability	codeword
A	1/2	0
B	1/4	10
C	1/8	110
D	1/8	111

Shannon (1948), §§6,9–12; original course examples

WORKED EXAMPLE

Entropy

The code length equals entropy in this example

COURSE-CREATED WORKED EXAMPLE

The same four probabilities give ideal lengths 1, 2, 3, 3.

$$H(p) = \tfrac{1}{2}\log_2 2 + \tfrac{1}{4}\log_2 4 + \tfrac{1}{4}\log_2 8 = 1.75$$

`p` (1/2, 1/4, 1/8, 1/8)

`H(p)` average surprise in bits

symbol	probability	codeword
A	1/2	0
B	1/4	10
C	1/8	110
D	1/8	111

Shannon (1948), §§6,9–12; original course examples

YOUR TURN

Entropy

Which binary source is hardest to predict?

Compare A: p=(1,0), B: p=(3/4,1/4), C: p=(1/2,1/2).

A. A has the largest entropy

B. B has the largest entropy

C. C has the largest entropy

Goodfellow et al., Deep Learning (2016), §3.13

DEFINITION

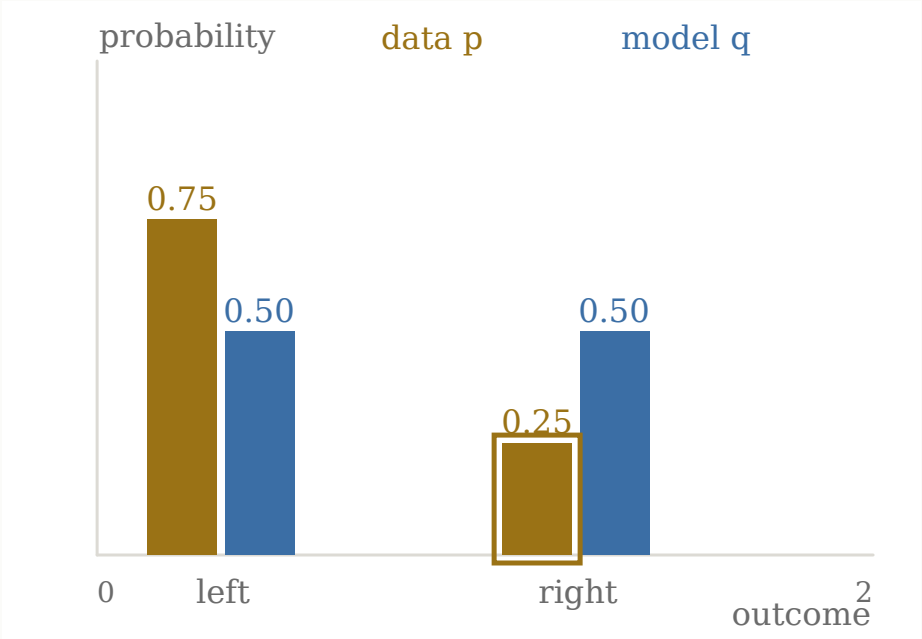
Cross entropy

The source and the prediction can differ

Outcomes follow p. The model scores them with q. Require q(x)>0 whenever p(x)>0.

$$H(p, q) = -\sum_{x} p(x) \log_2 q(x)$$

- `p` data-generating probabilities
- `q` model probabilities
- `H(p,q)` cross entropy in bits



Goodfellow et al., Deep Learning (2016), §3.13

DEFINITION

KL divergence

The excess cost of using the wrong probabilities

Subtract the irreducible entropy of the source from its cross entropy.

$$D_{\mathrm{KL}}(p\|q) = \sum_{x} p(x) \log_2 \frac{p(x)}{q(x)}$$

`p,q` source and model on the same outcomes

`D_KL` Kullback–Leibler divergence in bits

Goodfellow et al., Deep Learning (2016), §3.13

DERIVATION

KL divergence

The cross-entropy identity follows by expansion

Separate the logarithm of a ratio into a difference.

$$H(p, q) = H(p) + D_{\mathrm{KL}}(p \| q)$$

`H(p)` source uncertainty

`D_KL` extra cost of q

`H(p,q)` total average log penalty

Goodfellow et al., Deep Learning (2016), §3.13

KL divergence

The cross-entropy identity follows by expansion

Expand the log ratio

p is the fixed source; q is positive wherever p is positive.

1. $D_{\mathrm{KL}}(p\|q) = \sum_x p(x) \log_2 [p(x)/q(x)]$.
2. $= \sum_x p(x) \log_2 p(x) - \sum_x p(x) \log_2 q(x)$ by the log-ratio rule.

Goodfellow et al., Deep Learning (2016), §3.13

KL divergence

The cross-entropy identity follows by expansion

Identify the two entropy terms

H(p) is source entropy; H(p,q) is cross entropy.

1. $\sum_x p(x) \log_2 p(x) = -H(p)$.
2. $-\sum_x p(x) \log_2 q(x) = H(p, q)$.
3. Hence $D_{\mathrm{KL}}(p \| q) = H(p, q) - H(p)$, or $H(p, q) = H(p) + D_{\mathrm{KL}}(p \| q)$.

Goodfellow et al., Deep Learning (2016), §3.13

WORKED EXAMPLE

KL divergence

A two-route model has a measurable penalty

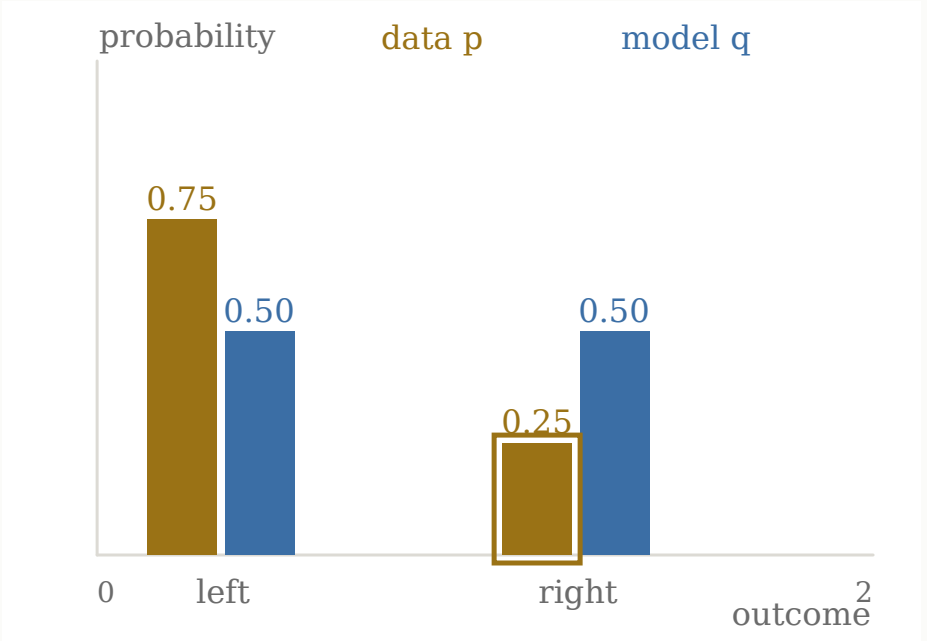
COURSE-CREATED WORKED EXAMPLE

Source p=(3/4,1/4). Model q=(1/2,1/2).

$$H(p,q) = 1, \quad H(p) \approx 0.811, \quad D_{\mathrm{KL}} \approx 0.189$$

`p,q` route probabilities in left/right order

`values` bits per observed route



[Goodfellow et al., Deep Learning (2016), §3.13]

YOUR TURN

KL divergence

What changes when source and model swap?

Now p=(1/2,1/2), q=(3/4,1/4). Find KL in bits.

A About 0.189

B About 0.208

C Exactly zero

Goodfellow et al., Deep Learning (2016), §3.13

DERIVATION

Cross entropy

Maximum likelihood uses the same average penalty

Assume independent observations $x_1,\dots,x_n$ from one model $q\theta$.

$$-\frac{1}{n}\log\prod_{i=1}^{n} q_\theta(x_i) = -\frac{1}{n}\sum_{i=1}^{n}\log q_\theta(x_i)$$

`θ` adjustable model parameters

`n,x_i` fixed sample count and observed values

Goodfellow et al., Deep Learning (2016), §5.5

Cross entropy

Maximum likelihood uses the same average penalty

Turn a likelihood product into a sum

$x_1,\ldots,x_n$ are observed iid data; θ is adjustable; log is natural.

1. $L(\theta) = \prod_{i=1}^{n} q_\theta(x_i)$ by the iid model.
2. $-n^{-1} \log L(\theta) = -n^{-1} \sum_i \log q_\theta(x_i)$ by the log-product rule.

Goodfellow et al., Deep Learning (2016), §5.5

Cross entropy

Maximum likelihood uses the same average penalty

Group repeated outcomes

N_x is the count of outcome x; empirical frequency is $\hat{p}_n(x)=N_x/n$.

1. $-n^{-1}\sum_i \log q_\theta(x_i) = -\sum_x (N_x/n)\log q_\theta(x)$.
2. $= -\sum_x \hat{p}_n(x)\log q_\theta(x)$, the empirical cross entropy.
3. The minus sign converts maximization to minimization; fixed 1/n does not change the optimizer.

Goodfellow et al., Deep Learning (2016), §5.5

YOUR TURN

KL divergence

Can a model forbid an event that occurs?

p=(0.9,0.1), q=(1,0). The second outcome is possible under p.

A KL(p||q) is infinite

B KL(p||q) is zero

C The zero-probability term is ignored

Goodfellow et al., Deep Learning (2016), §3.13

DERIVATION

KL divergence

Why the total cannot be negative

For positive u, $-\log u \geq 1-u$. Apply it where p(x)>0.

$$D_{\mathrm{KL}}(p\|q) \geq \sum_{p(x)>0} [p(x) - q(x)] \geq 0$$

- `u` q(x)/p(x)
- `log` natural logarithm here; base changes only scale

Goodfellow et al., Deep Learning (2016), §3.13

KL divergence

Why the total cannot be negative

Apply the tangent inequality

Use natural logs; u=q(x)/p(x)>0 on the support of p.

1. $-\log u \geq 1 - u$, with equality only at u=1.
2. $p(x)\log[p(x)/q(x)] \geq p(x) - q(x)$ after multiplying by p(x)>0.

Goodfellow et al., Deep Learning (2016), §3.13

KL divergence

Why the total cannot be negative

Sum over source support

S={x:p(x)>0}; both p and q have total mass 1.

1. $D_{\mathrm{KL}}(p\|q) \geq \sum_{x\in S}(p(x)-q(x))$.
2. $= 1 - \sum_{x\in S} q(x) \geq 0$.
3. Equality requires q=p on S and no q mass outside S. A log-base change preserves the sign.

Goodfellow et al., Deep Learning (2016), §3.13

COUNTEREXAMPLE

KL divergence

Nonnegative does not make it a distance

COURSE-CREATED WORKED EXAMPLE

KL can be asymmetric and violate the triangle inequality.

$$D(p\|r) \approx 1.758$$

$$D(p\|q) + D(q\|r) \approx 0.368 + 0.511 < 1.758$$

`p,q,r` (0.9,0.1), (0.5,0.5), (0.1,0.9)

`D` KL in nats for this example

Goodfellow et al., Deep Learning (2016), §3.13

INTERPRETATION

KL fitting

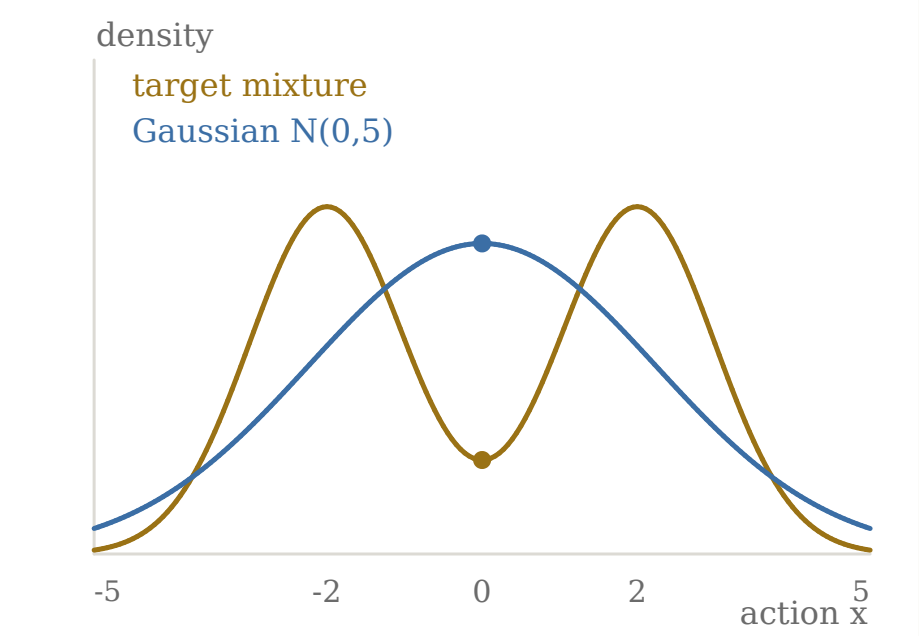
The direction changes what the model averages

p is fixed. qθ is restricted to a chosen model family.

$$\min_{\theta} D(p \| q_\theta) \quad \text{or} \quad \min_{\theta} D(q_\theta \| p)$$

`θ` adjustable model parameters

`p,q_theta` target distribution and approximation



Goodfellow et al., Deep Learning (2016), §3.13

DERIVATION

KL fitting

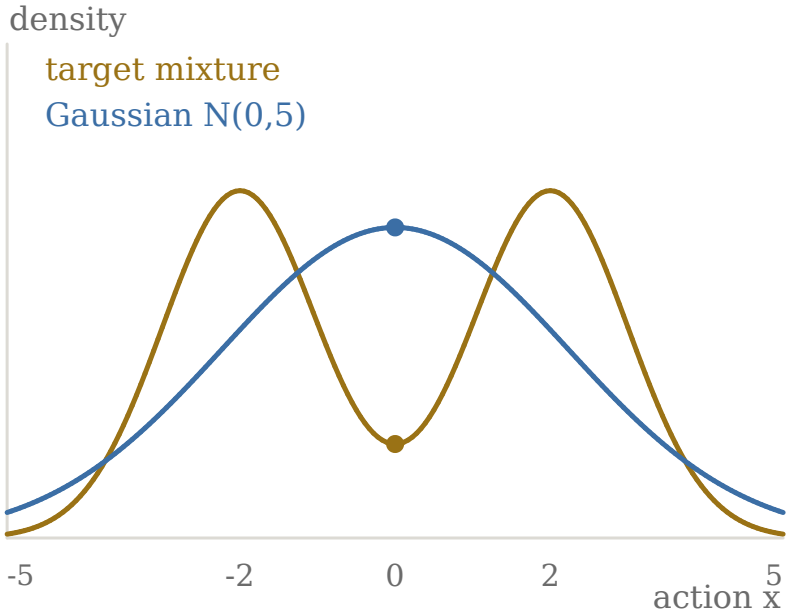
Forward KL within a Gaussian family

Continuous density p on R: finite second moment, finite differential entropy, variance v>0. Fit q=N(m,s²), s²>0.

$$-\mathbb{E}_p[\log q(X)] = \tfrac{1}{2}\log(2\pi s^2) + \frac{\mathbb{E}_p[(X-m)^2]}{2s^2}$$

`m,s²` adjustable Gaussian mean and variance

`mu,v` fixed target mean and variance



Goodfellow et al., Deep Learning (2016), §§3.8,3.9.3,3.13; course derivation

KL fitting

Forward KL within a Gaussian family

Insert the Gaussian model density

p is the fixed target density. q=N(m,s²), with adjustable m and s²>0.

1. $q(x) = (2\pi s^2)^{-1/2} \exp[-(x-m)^2/(2s^2)]$.
2. $-\log q(x) = \frac{1}{2}\log(2\pi s^2) + (x-m)^2/(2s^2)$.

Goodfellow et al., Deep Learning (2016), §§3.8,3.9.3,3.13; course derivation

KL fitting

Forward KL within a Gaussian family

Average under the continuous target

$\mu=\mathrm{E}_p[X]$, $v=\mathrm{Var}_p(X)>0$; differential entropy $h(p)=-\int p(x)\log p(x)\,dx$ is finite.

1. $C(m, s^2) = -\mathbb{E}_p[\log q(X)] = \frac{1}{2}\log(2\pi s^2) + \mathbb{E}_p[(X-m)^2]/(2s^2)$.
2. $D_{\mathrm{KL}}(p\|q) = \int p(x)\log[p(x)/q(x)]\,dx = C(m, s^2) - h(p)$.
3. Finite fixed h(p) makes minimizing KL equivalent to minimizing C.

Goodfellow et al., Deep Learning (2016), §§3.8,3.9.3,3.13; course derivation

DERIVATION

KL fitting

The best Gaussian center is the target mean

Expand around $\mu = E_p[X]$. The centered residual has zero expectation.

$$\mathbb{E}_p[(X-m)^2] = v + (\mu - m)^2$$

- v target variance $E[(X-\mu)^2]$
- m candidate Gaussian mean

Goodfellow et al., Deep Learning (2016), §§3.8,3.9.3,3.13; course derivation

KL fitting

The best Gaussian center is the target mean

Expand the square around the target mean

$\mu = E_p[X]$, $v = \mathrm{Var}_p(X)$, m is the adjustable Gaussian mean.

1. $X - m = (X - \mu) + (\mu - m).$
2. $(X - m)^2 = (X - \mu)^2 + 2(X - \mu)(\mu - m) + (\mu - m)^2.$

Goodfellow et al., Deep Learning (2016), §§3.8,3.9.3,3.13; course derivation

KL fitting

The best Gaussian center is the target mean

Average and minimize

μ−m is fixed when taking expectation over X.

1. $\mathbb{E}[X-\mu]=0$, so the cross term vanishes.
2. $\mathbb{E}[(X-m)^2]=v+(\mu-m)^2$.
3. For fixed $s^2>0$, the nonnegative square is minimized at m=μ.

Goodfellow et al., Deep Learning (2016), §§3.8,3.9.3,3.13; course derivation

DERIVATION

KL fitting

The best Gaussian variance matches the spread

At m=μ, set r=s²>0 and minimize C(r).

$$C(r) = \tfrac{1}{2}\log(2\pi r) + \frac{v}{2r}, \qquad C\prime(r) = \frac{r - v}{2r^2}$$

- r candidate Gaussian variance
- v positive target variance

Goodfellow et al., Deep Learning (2016), §§3.8,3.9.3,3.13; course derivation

KL fitting

The best Gaussian variance matches the spread

Differentiate with respect to the variance

r=s²>0 is adjustable; v>0 is the target variance.

1. $C(r) = \frac{1}{2}\log(2\pi r) + v/(2r).$
2. $C\prime(r) = 1/(2r) - v/(2r^2) = (r - v)/(2r^2).$

Goodfellow et al., Deep Learning (2016), §§3.8,3.9.3,3.13; course derivation

KL fitting

The best Gaussian variance matches the spread

Verify the minimum, not just stationarity

The denominator $2r^2$ is positive for $r>0$.

1. For $r<v$, $C'(r)<0$. For $r>v$, $C'(r)>0$.
2. C decreases toward v and increases after v, so $r=v$ is the global minimum.
3. The optimal Gaussian has mean μ and variance v.

Goodfellow et al., Deep Learning (2016), §§3.8,3.9.3,3.13; course derivation

INTEGRATED EXAMPLE

KL fitting

One Gaussian can place its mean between two peaks

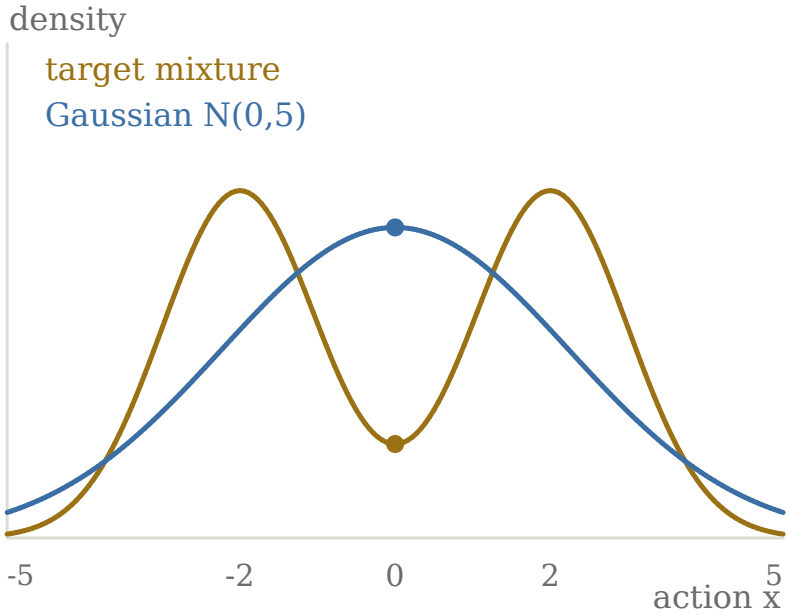
COURSE-CREATED WORKED EXAMPLE

Let p be an equal mixture of N(−2,1) and N(2,1).

$$\mu = 0, \qquad v = 1 + 4 = 5, \qquad q^* = \mathcal{N}(0, 5)$$

p two-component target density

q* best single Gaussian for forward KL



Goodfellow et al., Deep Learning (2016), §§3.8,3.9.3,3.13; course derivation

YOUR TURN

KL fitting

An uneven mixture shifts both fitted moments

$p=0.75\ \mathcal{N}(-2,1)+0.25\ \mathcal{N}(2,1)$. Fit one Gaussian by forward KL.

A. Mean -1, variance 1

B. Mean -1, variance 4

C. Mean 0, variance 5

Goodfellow et al., Deep Learning (2016), §§3.8,3.9.3,3.13; course derivation

INTERPRETATION

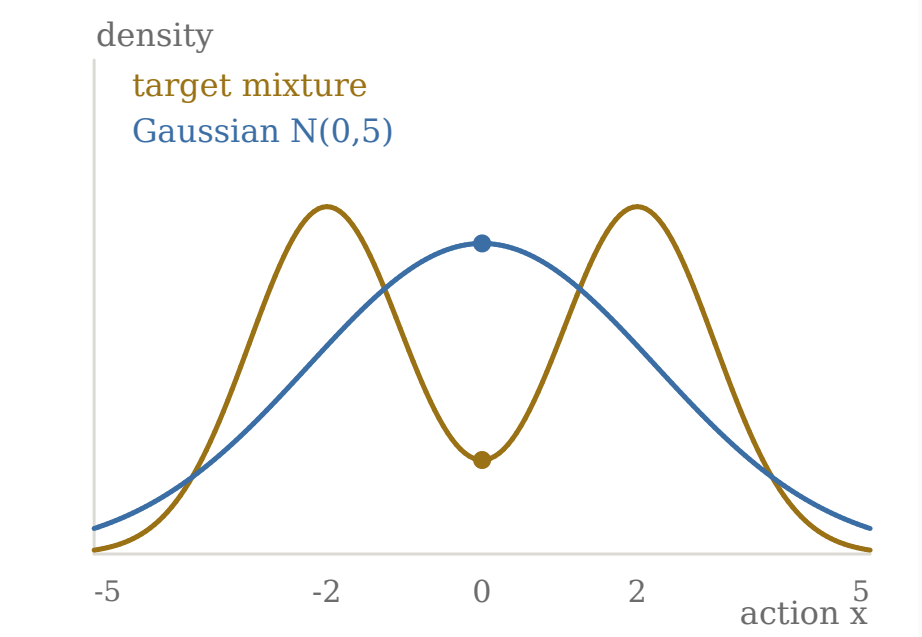
KL fitting

Reverse KL penalizes probability in a forbidden gap

If q(x)>0 where p(x)=0 on a region of positive q mass, reverse KL is infinite.

$$D(q\|p) = \mathbb{E}_q[\log q(X) - \log p(X)]$$

- p fixed target density
- q candidate model density



Goodfellow et al., Deep Learning (2016), §3.13

YOUR TURN

KL fitting

Does reverse KL always select one mode?

The candidate family contains the exact bimodal target p.

A Yes, one mode is always optimal

B No, q=p minimizes both directions

C Neither direction can reach zero

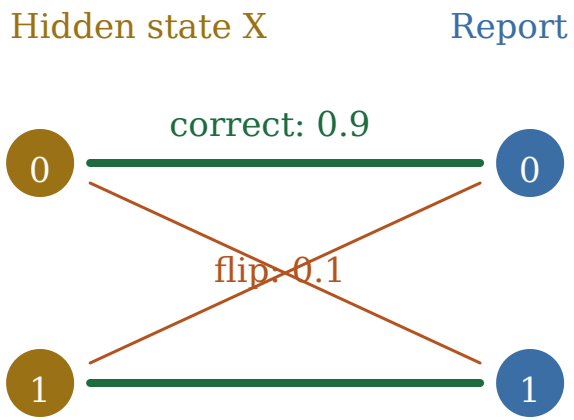
Goodfellow et al., Deep Learning (2016), §3.13

MOTIVATION

Mutual information

A sensor is useful when it changes our uncertainty

Let X be a hidden state and Y a sensor report. Does knowing Y help us predict X?



Shannon (1948), §§6,9–12; original course examples

WORKED EXAMPLE

Mutual information

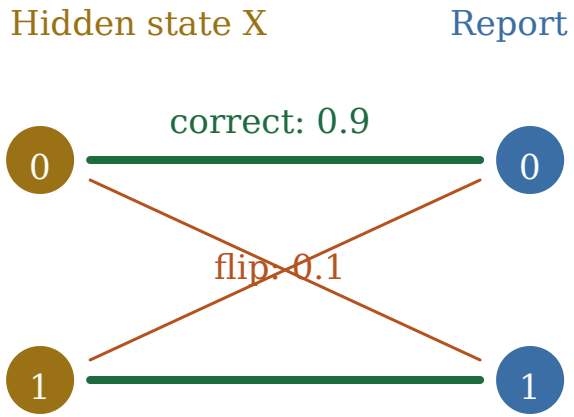
The marginals come from summing the joint table

COURSE-CREATED WORKED EXAMPLE

A fair binary state is reported correctly with probability 0.9.

$$p_{XY} = \begin{bmatrix} 0.45 & 0.05 \\ 0.05 & 0.45 \end{bmatrix}$$

`rows`	X=0,1
`columns`	Y=0,1
`p_X,p_Y`	row and column sums, each (1/2,1/2)



Shannon (1948), §§6,9–12; original course examples

DEFINITION

Mutual information

Compare dependence with the independent baseline

The joint and product-of-marginals distributions live on the same pairs (x,y).

$$I(X;Y) = \sum_{x,y} p(x,y) \log_2 \frac{p(x,y)}{p_X(x) p_Y(y)}$$

`I(X;Y)` mutual information in bits

`p_X,p_Y` marginal distributions

Shannon (1948), §§6,9–12; original course examples

DERIVATION

Mutual information

Information is expected uncertainty reduction

Use p(x,y)=p(y)p(x|y) inside the log ratio.

$$I(X;Y) = H(X) - H(X \mid Y)$$

`H(X|Y)` average entropy of X after observing Y

`H(X)` entropy before observing Y

Shannon (1948), §§6,9–12; original course examples

Mutual information

Information is expected uncertainty reduction

Cancel the report probability

p(x,y)=p_Y(y)p(x|y) on positive joint-probability pairs.

1. $\log_2 \frac{p(x,y)}{p_X(x)p_Y(y)} = \log_2 \frac{p(x|y)}{p_X(x)}.$
2. $= \log_2 p(x \mid y) - \log_2 p_X(x).$

Shannon (1948), §§6,9–12; original course examples

Mutual information

Information is expected uncertainty reduction

Average over the joint distribution

H(X|Y)=−Σₓ,ᵧp(x,y)log₂p(x|y) is conditional entropy.

1. $I(X;Y) = \sum_{x,y} p(x,y) \log_2 p(x \mid y) - \sum_{x,y} p(x,y) \log_2 p_X(x)$.
2. The first sum is −H(X|Y). Summing over y makes the second sum −H(X).
3. Therefore I(X;Y)=H(X)−H(X|Y), an expected reduction.

Shannon (1948), §§6,9–12; original course examples

INTEGRATED EXAMPLE

Mutual information

A noisy binary sensor supplies part of a bit

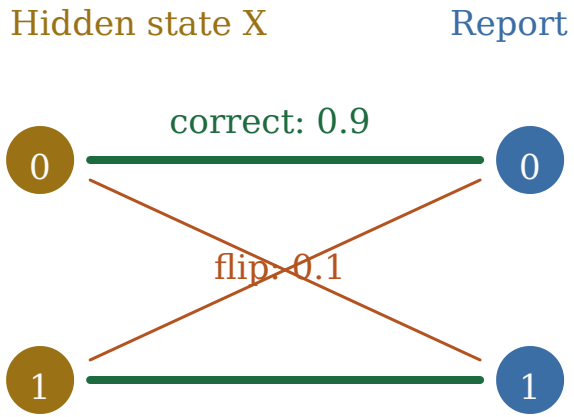
COURSE-CREATED WORKED EXAMPLE

X is fair. The sensor flips the state independently with probability e=0.1.

$$I(X;Y) = 1 - h_2(e) \approx 0.531$$

`h_2(e)` $-e \log_2 e - (1-e)\log_2(1-e)$

`e` independent binary flip probability



P(X=0) = P(X=1) = 0.5

Shannon (1948), §§6,9–12; original course examples

YOUR TURN

Mutual information

A noisier sensor leaves more uncertainty

X is fair. Y flips X independently with probability e=0.25. Find I(X;Y).

A About 0.811 bits

B About 0.189 bits

C Exactly 0.75 bits

Shannon (1948), §§6,9–12; original course examples

WORKED EXAMPLE

Mutual information

Zero covariance can hide a nonlinear dependence

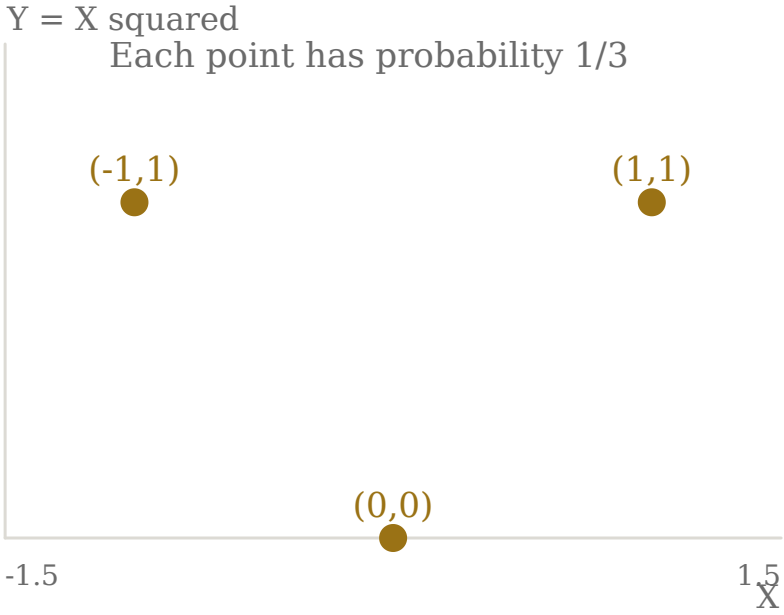
COURSE-CREATED WORKED EXAMPLE

X is uniform on $\{-1,0,1\}$. Set $Y=X^2$.

$$\mathrm{Cov}(X,Y) = \mathbb{E}[X^3] - \mathbb{E}[X]\mathbb{E}[X^2] = 0$$

`X,Y` random input and its square

`Cov` covariance; both variances are positive



Goodfellow et al., Deep Learning (2016), §3.8; original example

YOUR TURN

Mutual information

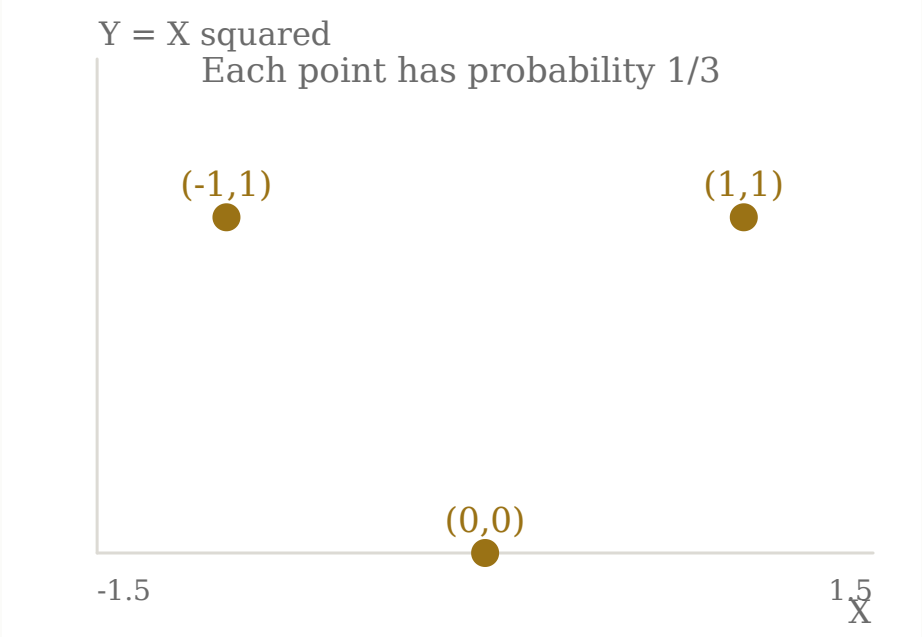
How much does the square reveal?

X is uniform on $\{-1,0,1\}$; $Y=X^2$. $H(Y|X)=0$.

A. $I(X;Y)=0$ because covariance is zero

B. $I(X;Y)=\log_2 3$

C. $I(X;Y)=h_2(1/3)\approx 0.918$ bits



Shannon (1948), §§6,9–12; original course examples

YOUR TURN

Mutual information

Compare a direct and an inverted sensor

X is a fair binary state. Sensor A gives $Y=X$. Sensor B gives $Y=1-X$.

A. A has more mutual information

B. Both provide 1 bit

C. B provides zero information

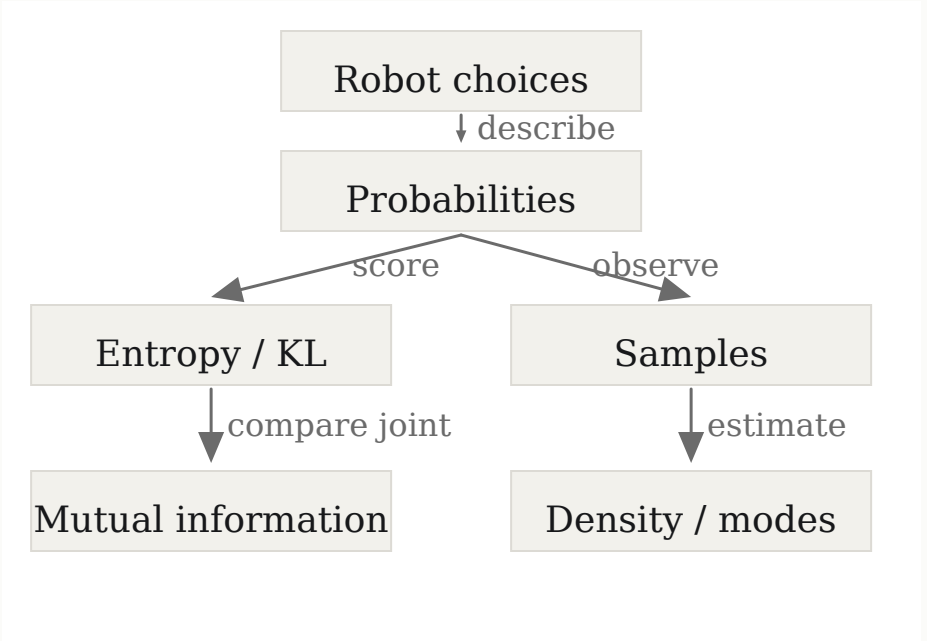
Shannon (1948), §§6,9–12; original course examples

SYNTHESIS

From distributions to data

The formulas need evidence from observations

Entropy compares uncertainty. KL compares models. Mutual information compares dependence. Samples estimate all three.



Goodfellow et al., Deep Learning (2016), §3.13

DEFINITION

Monte Carlo estimation

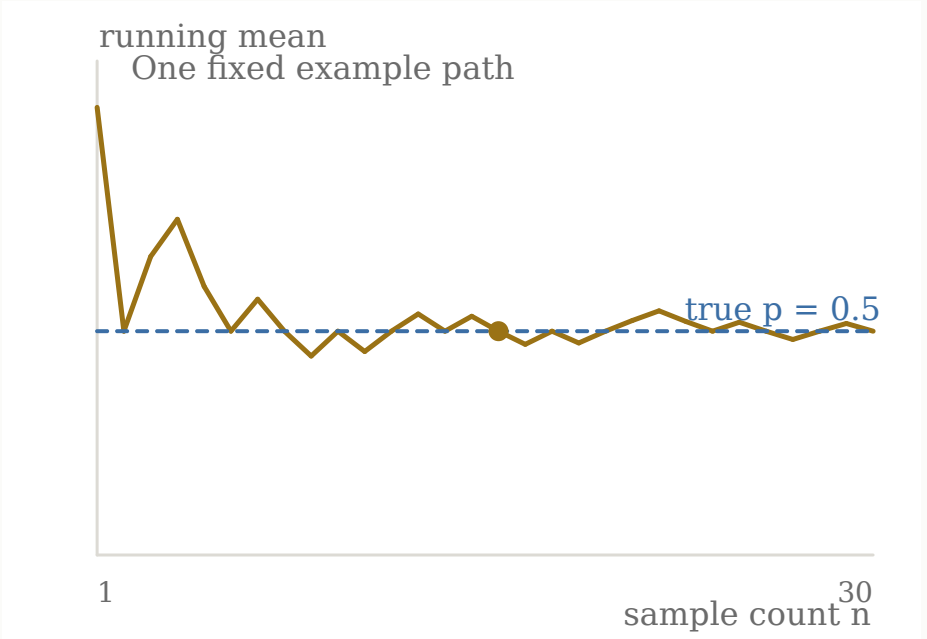
An average estimates a chosen expectation

Collect n independent and identically distributed draws from p, with n chosen in advance.

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} g(X_i), \qquad \mu = \mathbb{E}_p[g(X)]$$

`X_i` independent draws from the same distribution

`g,mu` measured quantity and its true mean



UC Berkeley CS70 (2017), Note18: mean, variance, concentration

DERIVATION

Monte Carlo estimation

Linearity explains unbiasedness

Assume E|g(X)| is finite and fixed n.

$$\mathbb{E}[\hat{\mu}_n] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[g(X_i)] = \mu$$

μˆ random sample average

μ common expectation of each sample value

UC Berkeley CS70 (2017), Note18: mean, variance, concentration

Monte Carlo estimation

Linearity explains unbiasedness

Use linearity of expectation

n is fixed; every measured value g(Xᵢ) has the same finite expectation μ.

1. $\mathbb{E}[\hat{\mu}_n] = \mathbb{E}[n^{-1} \sum_i g(X_i)]$.
2. $= n^{-1} \sum_i \mathbb{E}[g(X_i)]$ by linearity.
3. $= n^{-1}(n\mu) = \mu$ by the common sample expectation.

[UC Berkeley CS70 (2017), Note18: mean, variance, concentration]

DERIVATION

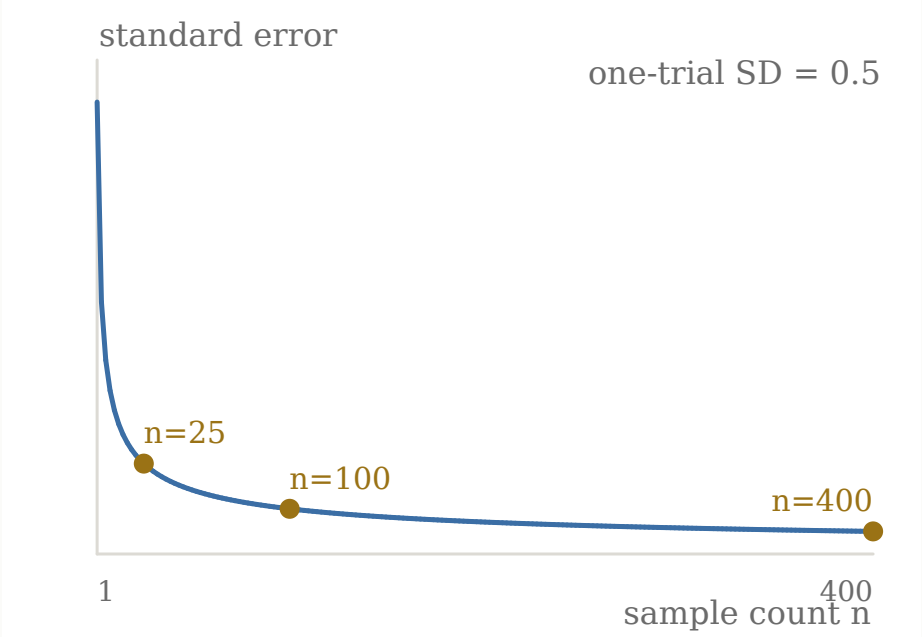
Monte Carlo estimation

Independence makes the variance shrink

Assume $\mathrm{Var}(g(X))=\sigma^2<\infty$ and independent samples.

$$\mathrm{Var}(\hat{\mu}_n) = \frac{\sigma^2}{n}, \qquad \mathrm{SE}(\hat{\mu}_n) = \frac{\sigma}{\sqrt{n}}$$

- σ^2 variance of one measured value
- SE standard deviation of the estimator



UC Berkeley CS70 (2017), Note18: mean, variance, concentration

Monte Carlo estimation

Independence makes the variance shrink

Expand the variance of the average

$Z_i = g(X_i)$ are independent with common finite variance σ^2.

1. $\operatorname{Var}(\hat{\mu}_n) = n^{-2} \operatorname{Var}(\sum_i Z_i)$.
2. $= n^{-2} \sum_i \operatorname{Var}(Z_i)$ because all cross covariances vanish.
3. $= n^{-2}(n\sigma^2) = \sigma^2/n$.

[UC Berkeley CS70 (2017), Note18: mean, variance, concentration]

Monte Carlo estimation

Independence makes the variance shrink

Convert variance to standard error

Standard error is the standard deviation of the estimator.

1. $\mathrm{SE}(\hat{\mu}_n) = \sqrt{\mathrm{Var}(\hat{\mu}_n)} = \sigma/\sqrt{n}$.
2. Four times as many independent samples gives half the standard error.
3. This describes the average, not the spread of one observation.

UC Berkeley CS70 (2017), Note18: mean, variance, concentration

YOUR TURN

Monte Carlo estimation

What does four times as much data buy?

Independent draws have the same finite variance. Increase n from 100 to 400.

A. Standard error becomes one quarter

B. Standard error becomes one half

C. Every realized estimate is closer

UC Berkeley CS70 (2017), Note18: mean, variance, concentration

DERIVATION

Monte Carlo estimation

A variance bound gives a probability statement

For fixed n and ε>0, the squared error is at least ε² on the large-error event.

$$P(|\hat{\mu}_n - \mu| \geq \varepsilon) \leq \frac{\sigma^2}{n\varepsilon^2}$$

`epsilon` chosen error tolerance

`σ²` finite variance of g(X)

[UC Berkeley CS70 (2017), Note18: mean, variance, concentration]

Monte Carlo estimation

A variance bound gives a probability statement

Bound squared error on the bad event

$E=|\hat{\mu}_n-\mu|$, $\varepsilon>0$, and $A=\{E\geq\varepsilon\}$.

1. On A, $E^2\geq\varepsilon^2$; outside A, $E^2\geq 0$.
2. Therefore $\mathbb{E}[E^2] \geq \varepsilon^2 P(A)$.

[UC Berkeley CS70 (2017), Note18: mean, variance, concentration]

Monte Carlo estimation

A variance bound gives a probability statement

Insert the estimator variance

The fixed-n iid average is unbiased and has variance σ²/n.

1. $\mathbb{E}[E^2] = \mathrm{Var}(\hat{\mu}_n) = \sigma^2/n.$
2. Dividing gives $P(|\hat{\mu}_n - \mu| \geq \varepsilon) \leq \sigma^2/(n\varepsilon^2).$
3. The result is an upper bound, not an equality or a guarantee for each dataset.

UC Berkeley CS70 (2017), Note18: mean, variance, concentration

INTEGRATED EXAMPLE

Monte Carlo estimation

A conservative trial count for a success rate

COURSE-CREATED WORKED EXAMPLE

Bernoulli success has variance $p(1-p) \le 1/4$. Target error 0.1 with failure probability ≤ 0.05.

$$\frac{1}{4n(0.1)^2} \le 0.05 \quad \Longrightarrow \quad n \ge 500$$

- p unknown success probability
- n fixed number of independent trials

UC Berkeley CS70 (2017), Note18: mean, variance, concentration

YOUR TURN

Monte Carlo estimation

A tighter tolerance changes sample cost

Use $p(1-p) \le 1/4$ and the variance bound. Keep failure probability ≤ 0.05; reduce error tolerance from 0.1 to 0.05.

A 250 trials suffice by this bound

B 1000 trials suffice by this bound

C 2000 trials suffice by this bound

[UC Berkeley CS70 (2017), Note18: mean, variance, concentration]

RESULT

Monte Carlo estimation

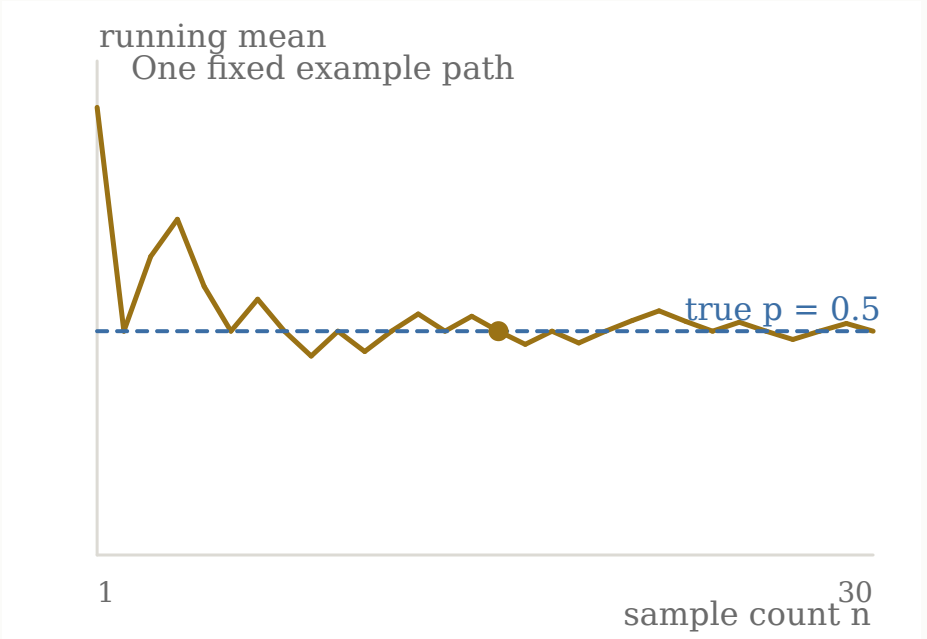
The law of large numbers is not a monotone path

For every fixed ε>0, the variance bound tends to zero as n grows.

$$P(|\hat{\mu}_n - \mu| \geq \varepsilon) \longrightarrow 0$$

$\hat{\mu}_n$ average of n iid measurements

μ finite-variance population mean



UC Berkeley CS70 (2017), Note18: mean, variance, concentration

BOUNDARY CASE

Monte Carlo estimation

Correlated repetitions are not fresh evidence

For general samples, the covariance terms remain.

$$\operatorname{Var}(\bar{Z}) = \frac{1}{n^2}\left[\sum_i \operatorname{Var}(Z_i) + 2\sum_{i<j} \operatorname{Cov}(Z_i, Z_j)\right]$$

`Z_i` measured values g(X_i)

`Z-bar` sample average

[UC Berkeley CS70 (2017), Note18: mean, variance, concentration]

DEFINITION

Stopping rules

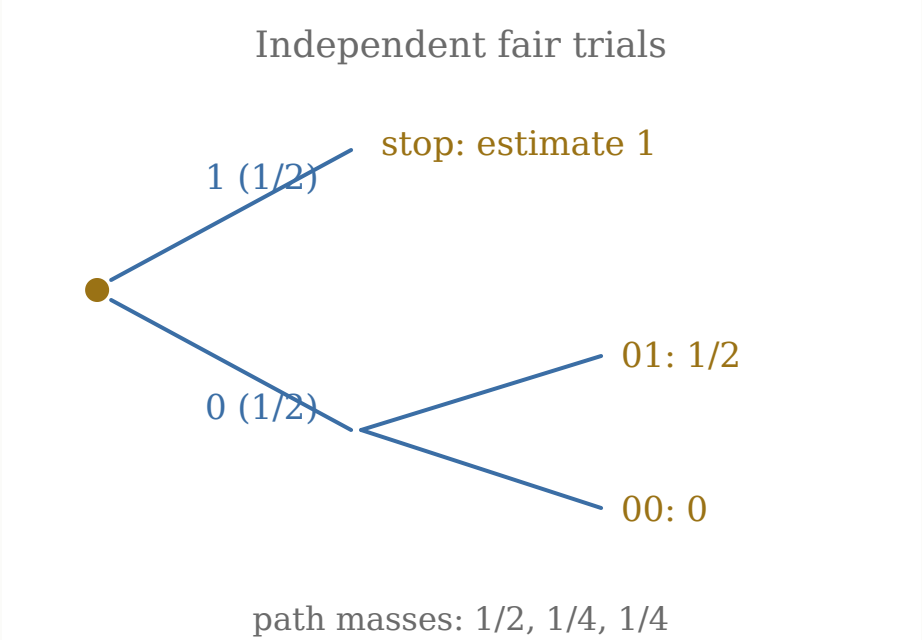
The data can determine when collection ends

A stopping time T decides whether to stop using observations available so far.

$$T = \begin{cases} 1, & X_1 = 1 \\ 2, & X_1 = 0 \end{cases}$$

`X_i` independent fair Bernoulli trials

`T` random sample count chosen by the rule



Sheffield, MIT 18.175 (2016), Lecture 14, pp.34–35; original example

WORKED EXAMPLE

Stopping rules

Enumerating paths exposes the selection effect

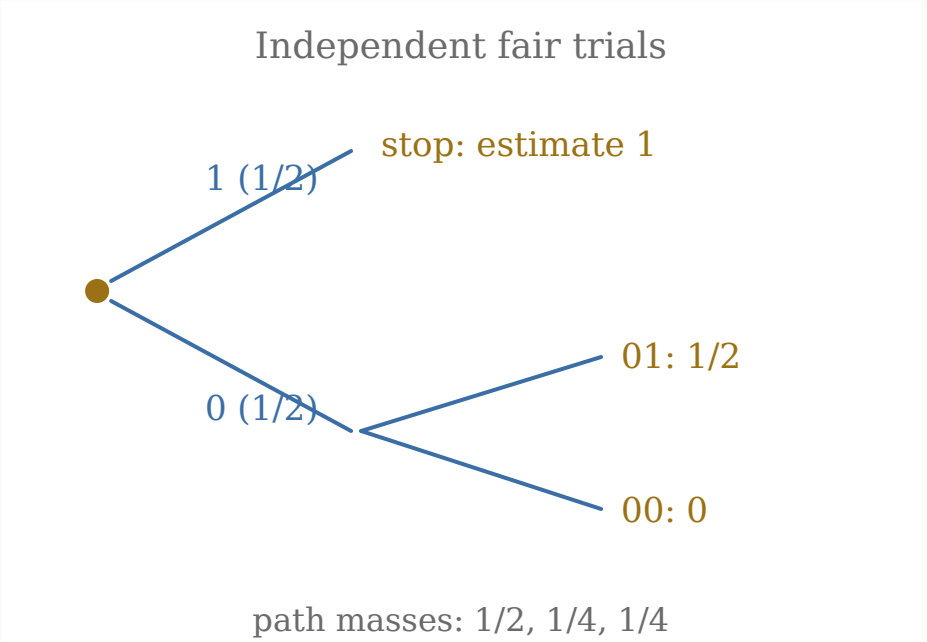
COURSE-CREATED WORKED EXAMPLE

Stop after first success, but take at most two trials. Each trial succeeds with probability 1/2.

$$(T, \hat{p}_T) = (1, 1), (2, 1/2), (2, 0)$$

`path probabilities` 1/2, 1/4, 1/4 respectively

`p̂T` success fraction at the stopping time



Sheffield, MIT 18.175 (2016), Lecture 14, pp.34–35; original example

YOUR TURN

Stopping rules

Is the stopped success fraction unbiased?

Completed paths: 1, 01, 00. Their probabilities are 1/2,1/4,1/4 and estimates 1,1/2,0.

A Its expectation is 1/2

B Its expectation is 5/8

C Its expectation is 1

Sheffield, MIT 18.175 (2016), Lecture 14, pp.34–35; original example

YOUR TURN

Stopping rules

Reverse the stopping preference

Stop after the first failure; otherwise take a second fair trial. Report success fraction.

A. Expected estimate 3/8

B. Expected estimate 1/2

C. Expected estimate 5/8

Sheffield, MIT 18.175 (2016), Lecture 14, pp.34–35; original example

ROBOTICS PAYOFF

Stopping rules

Match the guarantee to the collection rule

Fixed-n estimates need fixed-n evaluation. Adaptive collection needs an analysis designed for its stopping rule.

Howard et al. (2020), Introduction: time-uniform bounds

INTEGRATED EXAMPLE

Monte Carlo estimation

A rollout result needs an uncertainty statement

COURSE-CREATED WORKED EXAMPLE

An independent fixed evaluation has 80 successes in 100 trials. Use a Bernoulli model.

$$\hat{p} = 0.8, \qquad \widehat{\mathrm{SE}} = \sqrt{\frac{0.8(0.2)}{100}} = 0.04$$

$\hat{p}$	observed success fraction
estimated SE	plug-in standard error, not exact coverage

UC Berkeley CS70 (2017), Note18: mean, variance, concentration

YOUR TURN

Monte Carlo estimation

How many draws to see a rare route at least once?

A route has probability 0.1 in each independent draw. Find the smallest fixed integer n with detection probability at least 0.95.

$$P(\text{no rare route}) = (0.9)^n$$

n fixed independent sample count

A At least 10 draws

B At least 29 draws

C At least 95 draws

UC Berkeley CS70 (2017), Note18: mean, variance, concentration

MOTIVATION

Density and modes

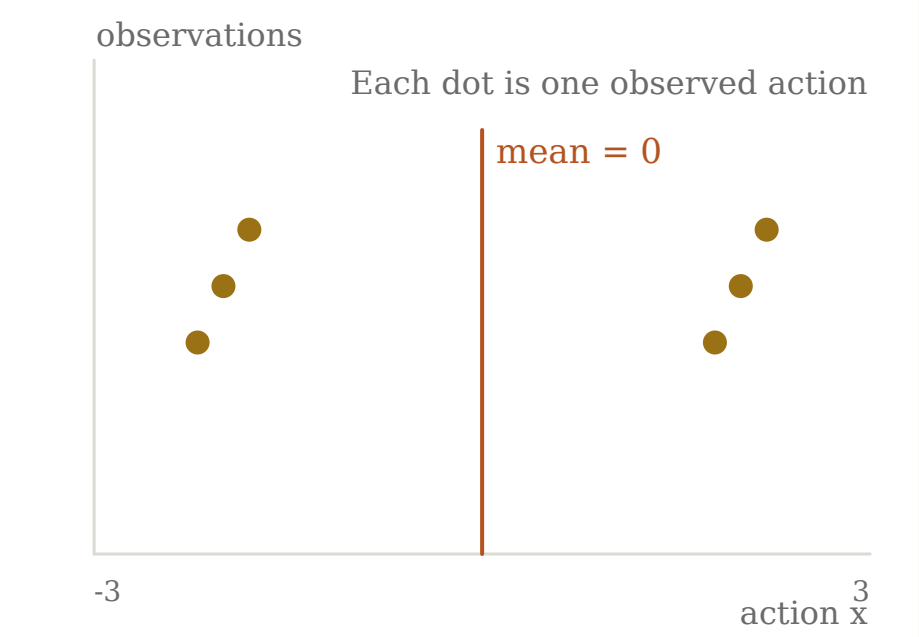
A mean can hide the choices a robot actually makes

Observed actions: -2.2, -2.0, -1.8, 1.8, 2.0, 2.2. Their mean is zero.

$$\bar{x} = 0, \qquad \text{dense groups lie near } -2 \text{ and } 2$$

`x_i` six observed scalar actions

`x̄` arithmetic mean



Wasserman, 36-708 Density Estimation, §§1,3–4

DEFINITION

Density and modes

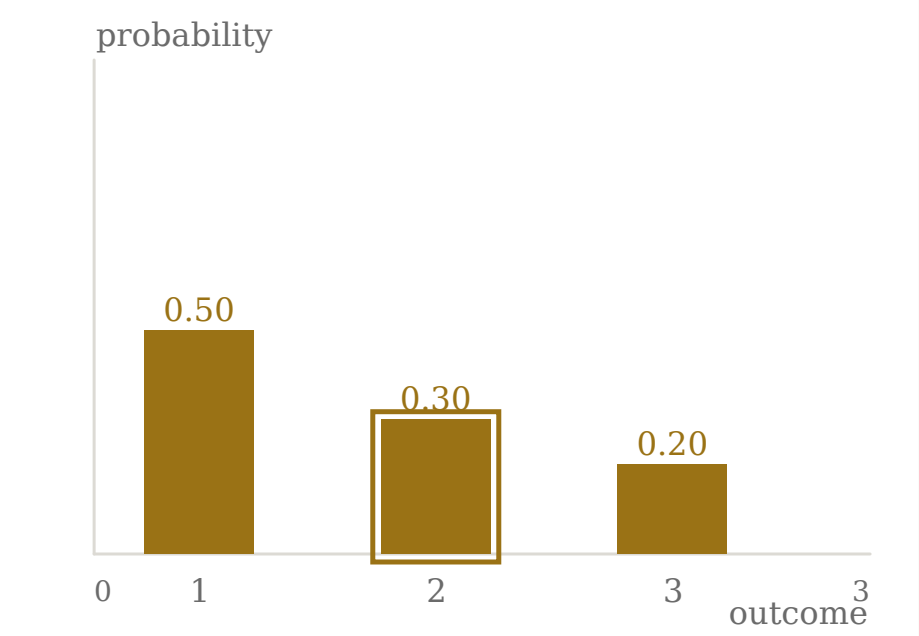
A discrete mode maximizes probability mass

For discrete outcomes, estimate each probability by its observed frequency.

$$\hat{p}_n(x) = \frac{N_x}{n}, \qquad \hat{m} \in \arg\max_x \hat{p}_n(x)$$

`N_x` number of observations equal to x

`m̂` one empirical mode; ties are possible



Wasserman, 36-708 Density Estimation, §§1,3–4

YOUR TURN

Density and modes

Does the largest observed count prove the true mode?

Three categories have counts 5,3,2 in ten independent observations.

A. The first category is the empirical mode

B. The first category is certainly the population mode

C. The sample mean defines the mode

Wasserman, 36-708 Density Estimation, §§1,3–4

RECALL

Density and modes

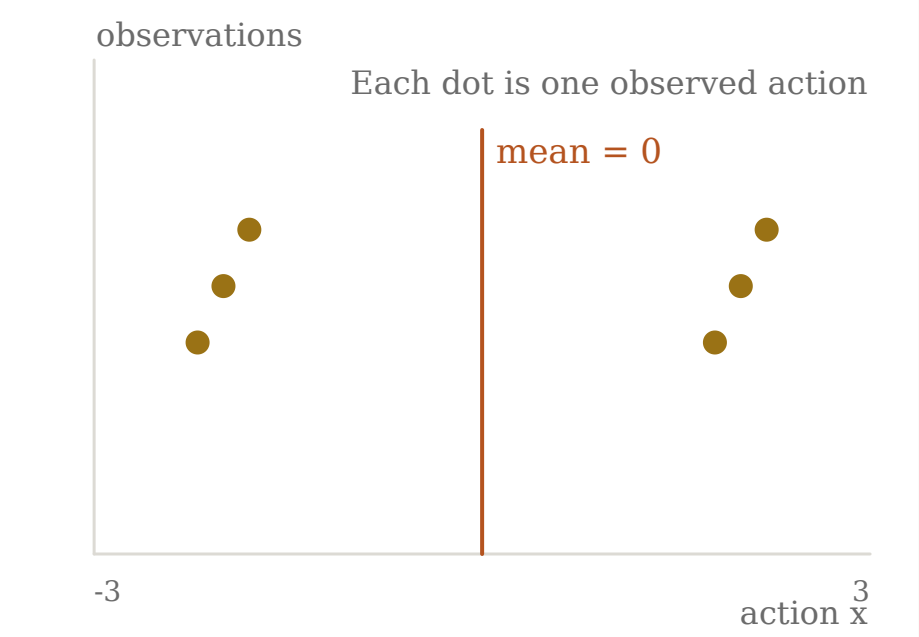
Continuous data rarely repeat exactly

For an absolutely continuous model, each exact value has probability zero, while intervals can have positive probability.

$$P(X = x) = 0, \qquad P(a \le X \le b) = \int_a^b p(u)\,du$$

`p(u)` density, not point probability

`a,b` interval endpoints



Goodfellow et al., Deep Learning (2016), §3.3.2

DEFINITION

Histograms

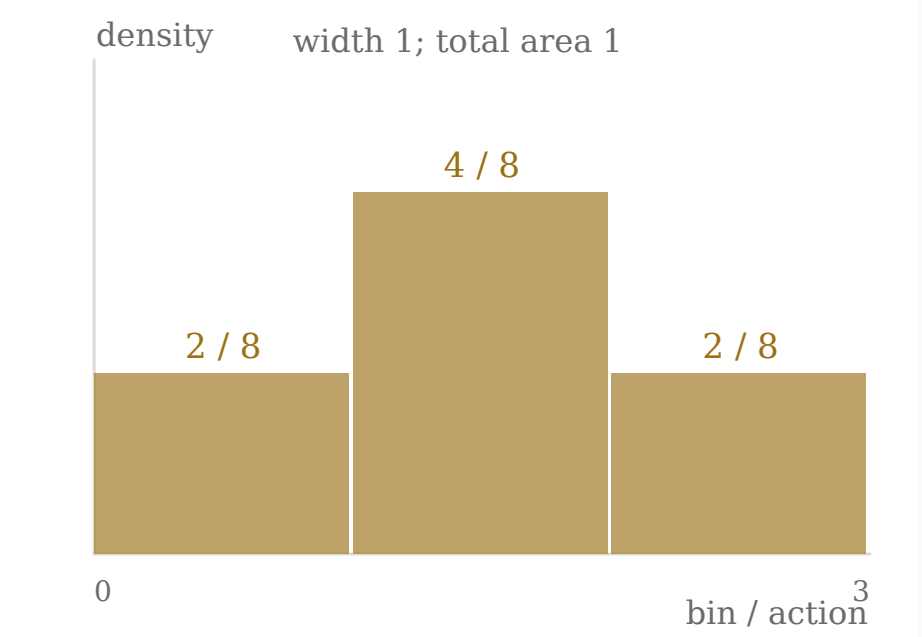
Counts become density after dividing by width

Choose non-overlapping bins with common width h. Count N_j observations in bin j.

$$\hat{p}(x) = \frac{N_j}{nh} \quad (x \text{ in bin } j)$$

`n,N_j` total count and bin count

`h` bin width in action units



Wasserman, 36-708 Density Estimation, §3; original example

WORKED EXAMPLE

Histograms

The area check catches normalization mistakes

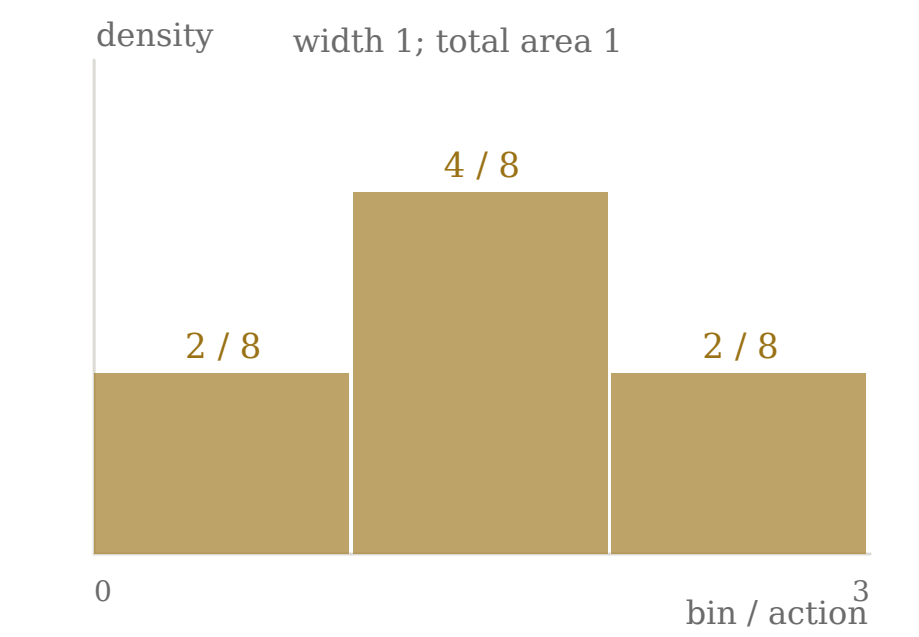
COURSE-CREATED WORKED EXAMPLE

Eight samples give counts 2,4,2 in three bins of width 1.

$$\hat{p} = (0.25, 0.5, 0.25), \qquad \sum_j h\hat{p}_j = 1$$

`h` common bin width 1

`p-hat_j` constant density in bin j



[Wasserman, 36-708 Density Estimation, §3; original example]

YOUR TURN

Histograms

Compare bins with different widths

n=7. Bin A: count 4,width 2. Bin B: count 3,width 1. Which has higher histogram density?

A Bin A

B Bin B

C They have equal density

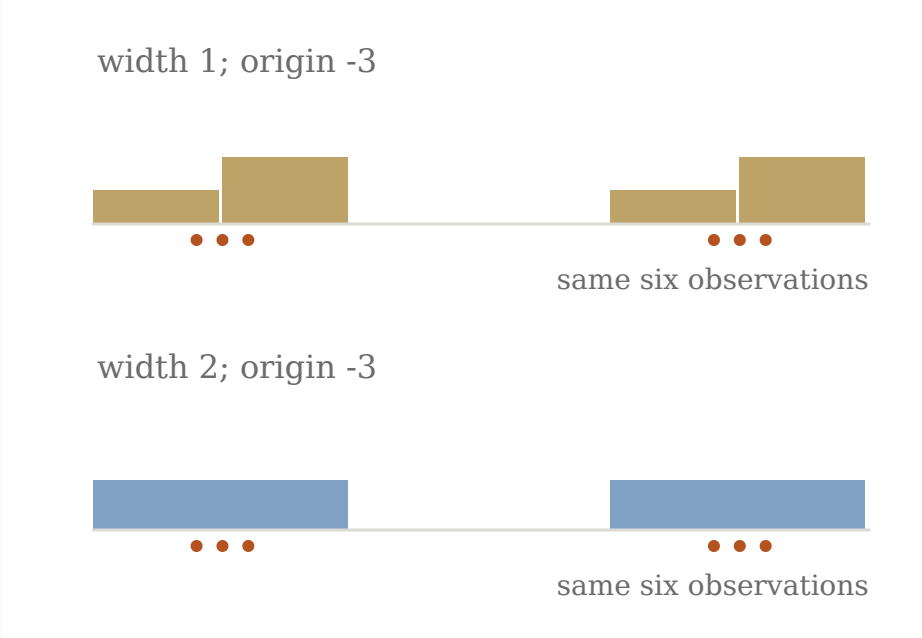
Wasserman, 36-708 Density Estimation, §3; original example

INTERPRETATION

Histograms

The apparent peak depends on the binning

Bin width controls averaging. Bin origin controls which observations share a bin.



Wasserman, 36-708 Density Estimation, §3; original example

DEFINITION

Kernel density estimation

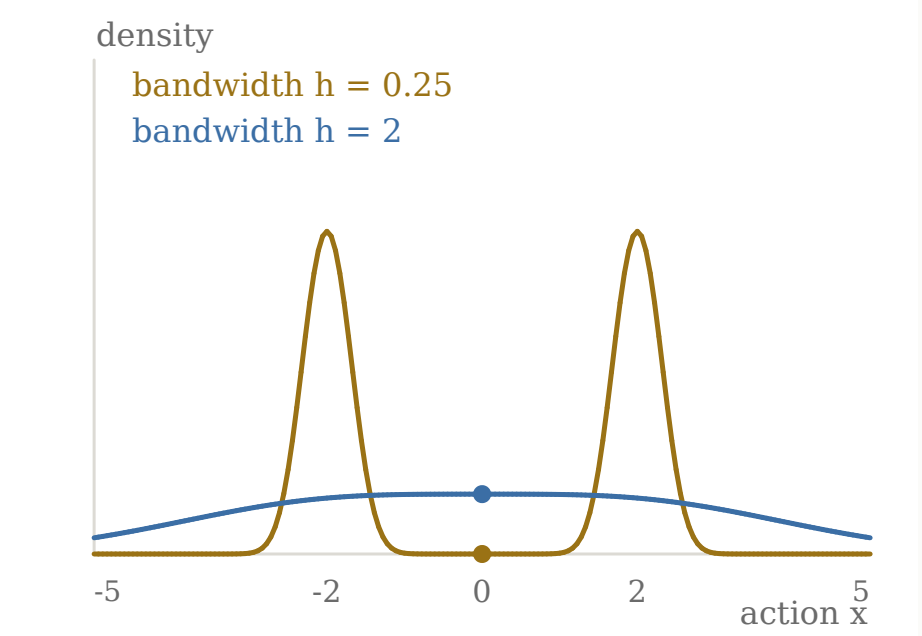
Each sample contributes a small smooth bump

Use a nonnegative kernel K with total integral 1 and a bandwidth h>0.

$$\hat{p}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h} \right)$$

`x_i,x` observed samples and query location

`h,K` bandwidth and unit-integral kernel



Wasserman, 36-708 Density Estimation, §4; original example

DERIVATION

Kernel density estimation

The normalization survives a change of scale

Substitute $u=(x-x_i)/h$, so $dx=h\,du$.

$$\int \frac{1}{h} K\left(\frac{x - x_i}{h}\right) dx = \int K(u)\, du = 1$$

- `h` positive bandwidth
- `x_i` fixed sample center
- `u` rescaled integration variable

Wasserman, 36-708 Density Estimation, §4; original example

Kernel density estimation

The normalization survives a change of scale

Change the integration variable

K has integral 1, h>0, and x_i is a fixed sample center.

1. Set $u = (x - x_i)/h$, so $dx = h\,du$.
2. $\int h^{-1} K((x - x_i)/h)\,dx = \int h^{-1} K(u) h\,du = 1.$

Wasserman, 36-708 Density Estimation, §4; original example

Kernel density estimation

The normalization survives a change of scale

Average the unit-area bumps

There are n kernel bumps and the estimator divides their sum by n.

1. $\int \hat{p}_h(x)\,dx = n^{-1}\sum_{i=1}^{n} 1 = 1.$
2. Without the 1/h factor, each bump would instead have area h.
3. Bandwidth changes width and height together, preserving total probability.

Wasserman, 36-708 Density Estimation, §4; original example

YOUR TURN

Kernel density estimation

What does a larger bandwidth change?

Use the same samples and a normalized Gaussian kernel. Increase h.

A. Wider bumps, same total area

B. More total probability mass

C. Narrower bumps and more sharp peaks

[Wasserman, 36-708 Density Estimation, §4; original example]

INTEGRATED EXAMPLE

Kernel density estimation

The same actions support different apparent shapes

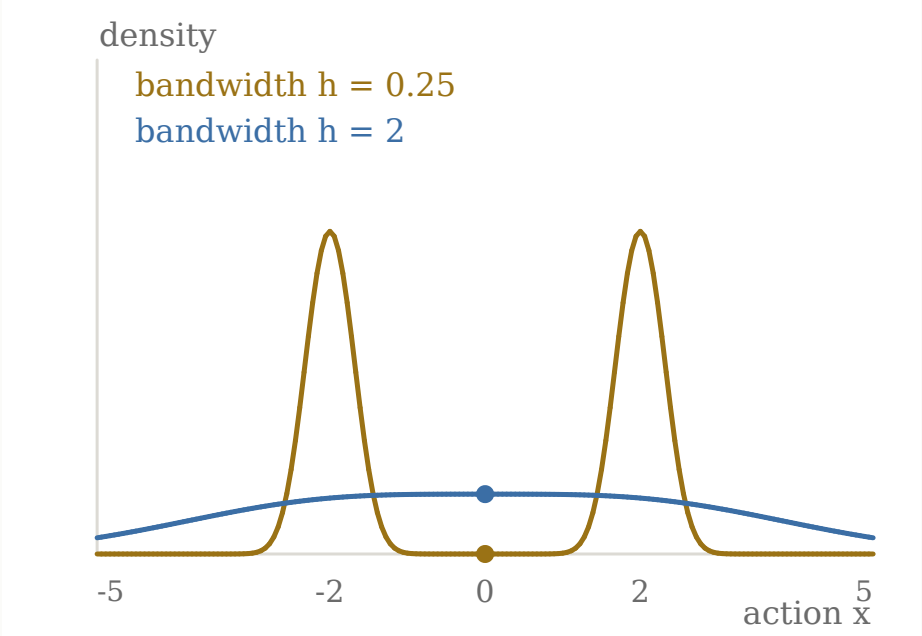
COURSE-CREATED WORKED EXAMPLE

Data: −2.2, −2.0, −1.8, 1.8, 2.0, 2.2. Compare Gaussian-kernel bandwidths 0.25 and 2.

$$\hat{m}_h \in \arg\max_x \hat{p}_h(x)$$

$\hat{m}_h$ estimated density mode, dependent on h

h bandwidth, in action units



Wasserman, 36-708 Density Estimation, §4; original example

YOUR TURN

Density and modes

Choose an action from two observed groups

Actions form two separated groups around -2 and 2. A robot needs one plausible action.

A. Always execute the mean 0

B. Model the distribution and check the sampled action

C. Declare every observed value a true mode

Wasserman, 36-708 Density Estimation, §§1,3–4

DEFINITION

Stable probability computation

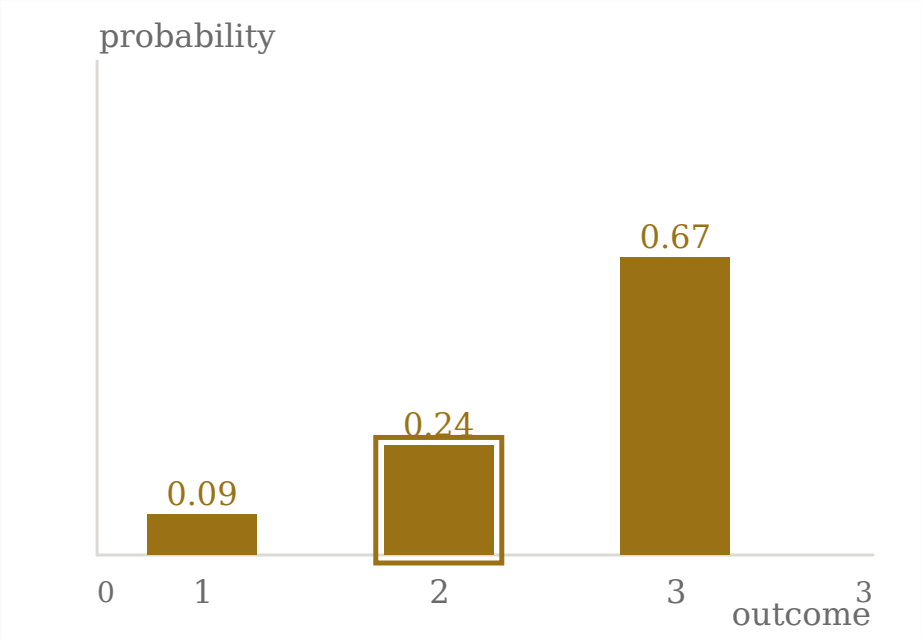
Exponentials turn scores into probabilities

A model produces finite real logits z₁,…,z_K. Normalize positive exponentials.

$$q_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

`z_i` unconstrained model score, called a logit

`q_i` normalized probability of class i



Blanchard, Higham & Higham (2021), §§1,4

DERIVATION

Stable probability computation

A shared shift leaves softmax unchanged

Multiply numerator and denominator by e^{-c}. Choose $c=\max_j z_j$.

$$q_i = \frac{e^{z_i - c}}{\sum_j e^{z_j - c}}$$

`c` largest finite logit

`z_i−c` nonpositive shifted logit

Blanchard, Higham & Higham (2021), §§1,4

Stable probability computation

A shared shift leaves softmax unchanged

Cancel a common exponential factor

z_i are finite logits; c is any finite shared constant.

1. $q_i = \frac{e^{z_i} e^{-c}}{\sum_j e^{z_j} e^{-c}}$.
2. $= \frac{e^{z_i - c}}{\sum_j e^{z_j - c}}$ by exponential multiplication.

Blanchard, Higham & Higham (2021), §§1,4

Stable probability computation

A shared shift leaves softmax unchanged

Choose the shift for numerical safety

Set $c=\max_j z_j$.

1. Every shifted score satisfies $z_i-c\le 0$.
2. Therefore $0<\exp(z_i-c)\le 1$ in exact arithmetic; at least one equals 1.
3. This avoids exponential overflow. Tiny terms can still underflow in finite precision.

Blanchard, Higham & Higham (2021), §§1,4

WORKED EXAMPLE

Stable probability computation

Large logits can encode ordinary probabilities

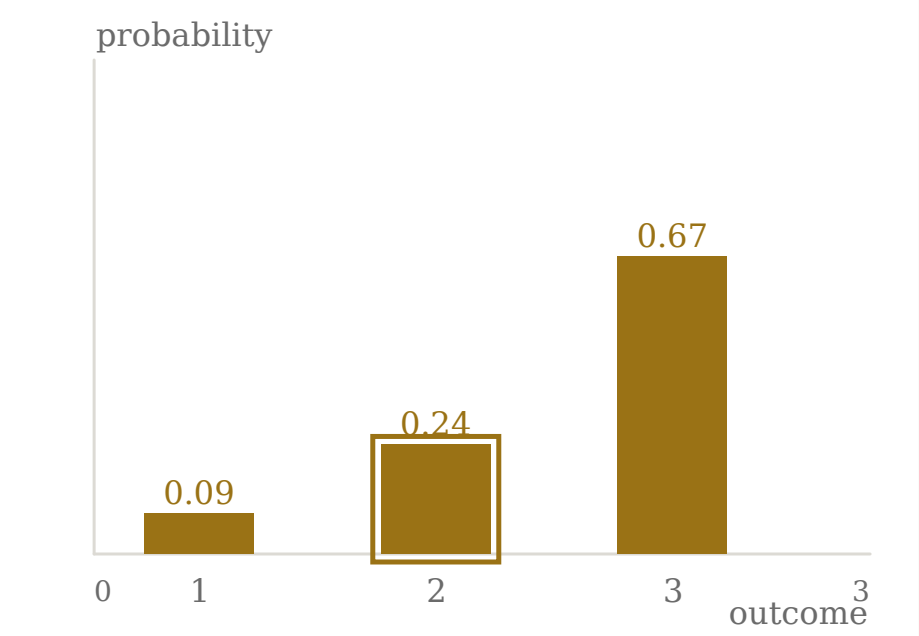
COURSE-CREATED WORKED EXAMPLE

Given z=(1000,1001,1002), subtract c=1002.

$$q = \frac{(e^{-2}, e^{-1}, 1)}{e^{-2} + e^{-1} + 1} \approx (0.090, 0.245, 0.665)$$

`z,q` logit vector and probability vector

`c` maximum logit 1002



Blanchard, Higham & Higham (2021), §§1,4

YOUR TURN

Stable probability computation

Which change preserves the distribution?

For finite logits z, compare two transformations before softmax.

A $z \to z + 1000 \cdot 1$

B $z \to 2z$

C $z \to -z$

Blanchard, Higham & Higham (2021), §§1,4

DERIVATION

Stable probability computation

Log-sum-exp keeps the normalization in log space

Factor the largest exponential before taking the logarithm.

$$\mathrm{LSE}(z) = c + \log \sum_j e^{z_j - c}, \quad c = \max_j z_j$$

`LSE(z)` log of the sum of exponentials

`c` maximum finite logit

Blanchard, Higham & Higham (2021), §§1,4

Stable probability computation

Log-sum-exp keeps the normalization in log space

Factor before taking the logarithm

$c = \max_j z_j$ for finite logits z.

1. $\sum_j e^{z_j} = e^c \sum_j e^{z_j - c}$.
2. $\log \sum_j e^{z_j} = \log e^c + \log \sum_j e^{z_j - c}$.
3. $\mathrm{LSE}(z) = c + \log \sum_j e^{z_j - c}$.

Blanchard, Higham & Higham (2021), §§1,4

Stable probability computation

Log-sum-exp keeps the normalization in log space

Track the constant

LSE is a log normalization value, not a probability ratio.

1. Adding c to every logit adds c to LSE.
2. The shift term cancels in softmax ratios but must be restored in LSE.
3. For z=(1000,1000), LSE=1000+log 2.

Blanchard, Higham & Higham (2021), §§1,4

DERIVATION

Stable probability computation

Compute log probabilities before exponentiating

For a known class y, use the log probability directly in cross entropy.

$$\log q_i = z_i - \mathrm{LSE}(z), \qquad \ell = -\log q_y$$

- y observed class index
- ℓ single-example negative log loss in nats

Blanchard, Higham & Higham (2021), §§1,4

Stable probability computation

Compute log probabilities before exponentiating

Take the log of the softmax ratio

q_i is the class probability and $\mathrm{LSE}(z)=\log\Sigma_j\exp(z_j)$.

1. $\log q_i = \log e^{z_i} - \log \sum_j e^{z_j} = z_i - \mathrm{LSE}(z)$.
2. For the observed class y, $\ell = -\log q_y$.

Blanchard, Higham & Higham (2021), §§1,4

Stable probability computation

Compute log probabilities before exponentiating

Keep the computation in shifted form

$c=\max_j z_j$ and y is the observed class index.

1. $\log q_i = (z_i - c) - \log \sum_j e^{z_j - c}$.
2. $\ell = \log \sum_j e^{z_j - c} - (z_y - c)$.
3. Average ℓ across observations to obtain empirical cross entropy in nats.

Blanchard, Higham & Higham (2021), §§1,4

YOUR TURN

Stable probability computation

Compute the observed-class log loss

z=(1000,999). The observed class is the second. Compute loss in nats.

A About 0.313

B About 1.313

C Infinite because e^{1000} overflows

Blanchard, Higham & Higham (2021), §§1,4

DERIVATION

Stable probability computation

The gradient of log-sum-exp is softmax

Differentiate the scalar normalizer with respect to each score.

$$\frac{\partial}{\partial z_i} \log \sum_j e^{z_j} = \frac{e^{z_i}}{\sum_j e^{z_j}} = q_i$$

`z`	column vector of logits
`gradient`	column probability vector q

Blanchard, Higham & Higham (2021), §§1,4

Stable probability computation

The gradient of log-sum-exp is softmax

Differentiate the scalar normalizer

$S=\Sigma_j \exp(z_j)$, and the logits are independent coordinates.

1. $\partial S/\partial z_i = e^{z_i}$.
2. $\partial \log S/\partial z_i = S^{-1}\partial S/\partial z_i = e^{z_i}/S = q_i$.

Blanchard, Higham & Higham (2021), §§1,4

Stable probability computation

The gradient of log-sum-exp is softmax

Differentiate the observed-class loss

ℓ=LSE(z)−z_y; e_y is the one-hot column target.

1. $\nabla_z \operatorname{LSE}(z) = q$.
2. $\nabla_z z_y = e_y$, so $\nabla_z \ell = q - e_y$.
3. The gradient components sum to 1−1=0, consistent with shared-shift invariance.

Blanchard, Higham & Higham (2021), §§1,4

YOUR TURN

Stable probability computation

The loss gradient changes the right scores

q=(0.7,0.2,0.1). The true class is 2. Use gradient descent on logits.

$$\nabla_z \ell = q - e_2$$

`e_2` one-hot target (0,1,0)

`z` adjustable logits

A Increase all logits equally

B Increase class 2, decrease classes 1 and 3

C Decrease class 2, increase classes 1 and 3

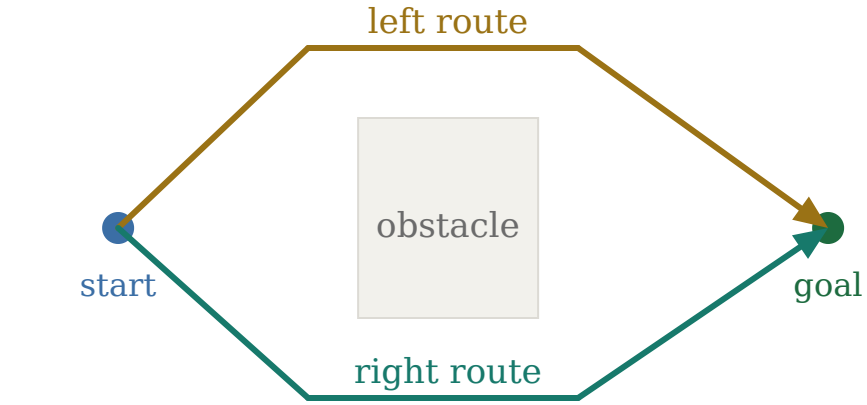
Blanchard, Higham & Higham (2021), §§1,4

INTEGRATED SETUP

Robot synthesis

Choose the quantities before calculating them

Routes follow p=(0.8,0.2); model q=(0.6,0.4).
Evaluate 100 independent preplanned trials.



Goodfellow et al., Deep Learning (2016), §3.13

INTEGRATED SOLUTION

Robot synthesis

Information and sampling answer different parts

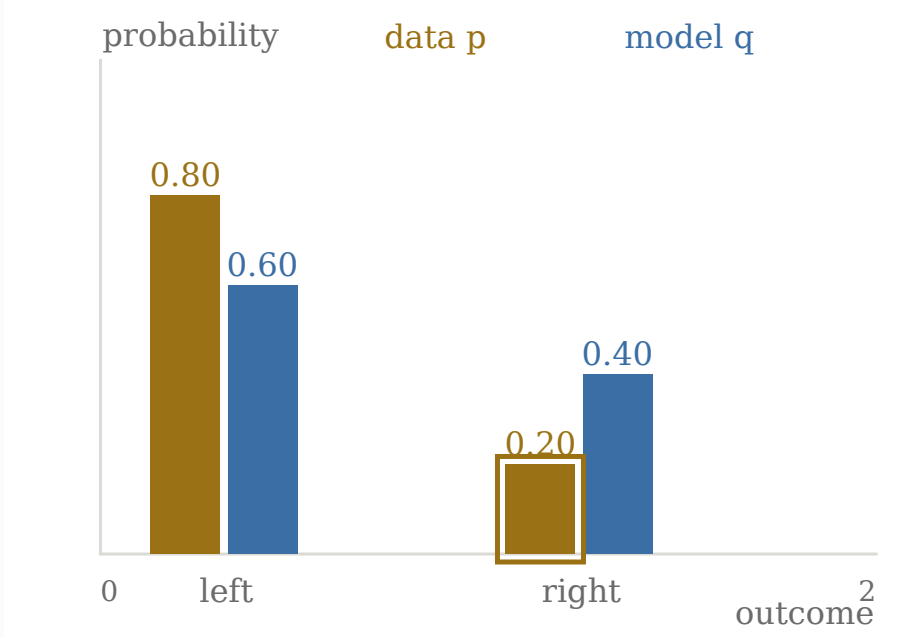
COURSE-CREATED WORKED EXAMPLE

p=(0.8,0.2), q=(0.6,0.4), n=100 independent trials.

$$H(p) \approx 0.722, \quad H(p,q) \approx 0.854, \quad D_{\mathrm{KL}} \approx 0.132$$

values bits per route; exact source probabilities given

SE sqrt(0.8·0.2/100)=0.04 for route frequency



Goodfellow et al., Deep Learning (2016), §3.13

YOUR TURN

Robot synthesis

Evaluate a changed route frequency

Source changes to p=(0.2,0.8); model stays q=(0.6,0.4). Collect a fresh fixed evaluation.

A Entropy stays the same; cross entropy rises

B Both quantities stay the same

C Entropy must increase

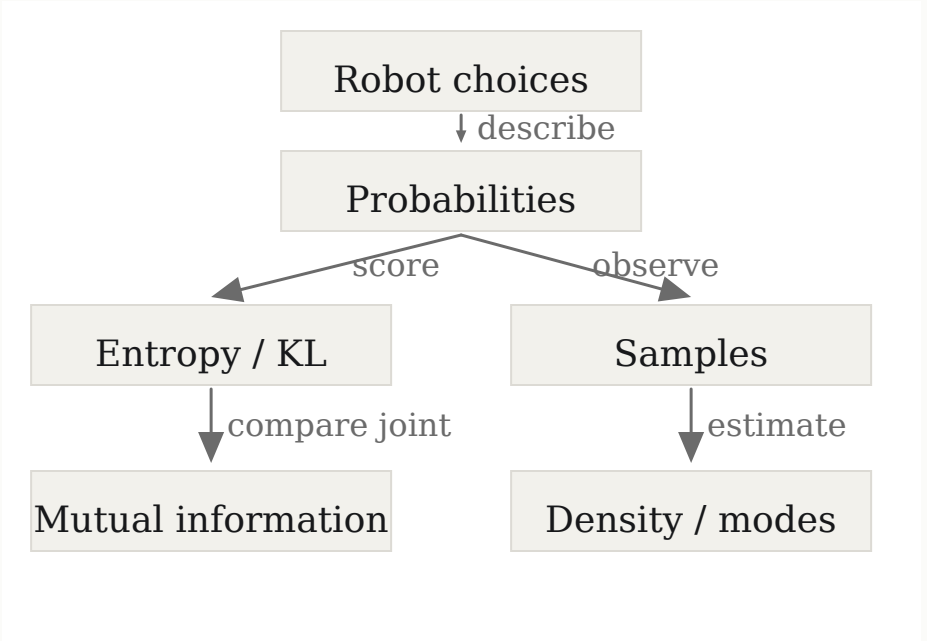
Goodfellow et al., Deep Learning (2016), §3.13

SYNTHESIS

The reasoning path

From scoring predictions to generating actions

Probabilities support log losses. Samples support estimates. Structured density models support multiple action choices.



Goodfellow et al., Deep Learning (2016), §3.13

SOURCES

References

Probability and information

1. Goodfellow, Bengio & Courville. Deep Learning. MIT Press, 2016. §§3.3,3.8,3.9,3.13,5.5.
2. Shannon. A Mathematical Theory of Communication. Bell System Technical Journal, 1948. §§6,9–12.
3. UC Berkeley. CS70 Note18: Chebyshev inequality and large numbers. 2017.

Original course examples and redraws. No source figures reproduced. Course materials: CC BY-NC 4.0.

SOURCES

References

Sampling, density and numerical methods

1. Sheffield. MIT 18.175, Lecture 14, 2016. pp.34–35.
2. Wasserman. Density Estimation. CMU 36-708. §§1,3–4.
3. Blanchard, Higham & Higham. Log-Sum-Exp and Softmax. IMA J. Numer. Anal., 2021. §§1,4.
4. Howard et al. Time-uniform Chernoff bounds. Probability Surveys 17, 2020.

Original course examples and redraws. No source figures reproduced. Course materials: CC BY-NC 4.0.