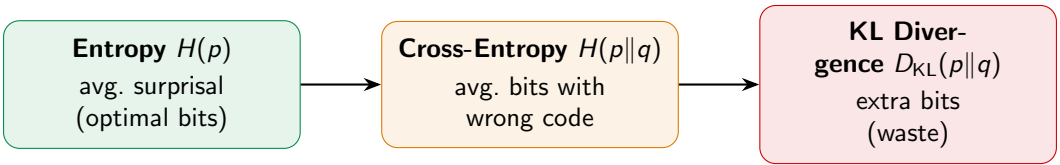
Information Theory II: ML Connections

KL = MLE · Cross-Entropy Loss · Forward/Reverse KL · Mutual Information

Recap: The Three Pillars



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

Today: We connect these concepts to machine learning.
KL $\rightarrow$ MLE, cross-entropy $\rightarrow$ log-loss, forward/reverse KL,
maximum entropy principle, and mutual information.

KL = Maximum Likelihood

The single most important connection between information theory and machine learning.

Minimizing KL = Maximizing Log-Likelihood

Data from true p, model family q_θ. Find θ making q_θ closest to p.

Step 1: Write out $D_{\mathrm{KL}}(p\|q_\theta)$:

$$D_{\mathrm{KL}}(p\|q_\theta) = \underbrace{\sum_x p(x)\log p(x)}_{=-H(p),\text{ doesn't depend on }\theta!} - \sum_x p(x)\log q_\theta(x)$$

Minimizing KL = Maximizing Log-Likelihood

Data from true p, model family q_θ. Find θ making q_θ closest to p.

Step 1: Write out $D_{\mathrm{KL}}(p\|q_\theta)$:

$$D_{\mathrm{KL}}(p\|q_\theta) = \underbrace{\sum_x p(x)\log p(x)}_{=-H(p),\ \text{doesn't depend on } \theta!} - \sum_x p(x)\log q_\theta(x)$$

Step 2: Drop the constant:

$$\arg\min_\theta \ D_{\mathrm{KL}}(p\|q_\theta) = \arg\max_\theta \sum_x p(x)\log q_\theta(x)$$

Minimizing KL = Maximizing Log-Likelihood

Data from true p, model family q_θ. Find θ making q_θ closest to p.

Step 1: Write out $D_{\mathrm{KL}}(p\|q_\theta)$:

$$D_{\mathrm{KL}}(p\|q_\theta) = \underbrace{\sum_x p(x)\log p(x)}_{=-H(p),\text{ doesn't depend on }\theta!} - \sum_x p(x)\log q_\theta(x)$$

Step 2: Drop the constant:

$$\arg\min_\theta \; D_{\mathrm{KL}}(p\|q_\theta) = \arg\max_\theta \sum_x p(x)\log q_\theta(x)$$

Step 3: Replace p by empirical $\hat{p}(x) = \frac{1}{n}\sum_{i=1}^n \mathbf{1}[x_i = x]$:

$$\arg\max_\theta \sum_x \hat{p}(x)\log q_\theta(x) = \arg\max_\theta \; \frac{1}{n}\sum_{i=1}^n \log q_\theta(x_i)$$

$$\arg\min_\theta \; D_{\mathrm{KL}}(p\|q_\theta) \;=\; \arg\max_\theta \sum_{i=1}^n \log q_\theta(x_i) \;=\; \textbf{MLE!}$$

Cross-Entropy Minimization = MLE

Since $H(p\|q_\theta) = H(p) + D_{\mathrm{KL}}(p\|q_\theta)$ and $H(p)$ is constant w.r.t. θ:

$$\arg\min_\theta \; D_{\mathrm{KL}}(p\|q_\theta) \;=\; \arg\min_\theta \; H(p\|q_\theta) \;=\; \arg\min_\theta \; -\sum_x p(x)\log q_\theta(x)$$

With empirical data:

$$\arg\min_\theta \; H(\hat{p}\|q_\theta) = \arg\min_\theta \left(-\frac{1}{n}\sum_{i=1}^{n} \log q_\theta(x_i) \right) = \text{MLE}$$

Minimize KL $=$ Minimize cross-entropy $=$ Maximize log-likelihood

All three are the same optimization problem!

This is why cross-entropy is THE standard loss function in ML.

Cross-Entropy Loss in Classification

Multi-class classification with g classes. For observation i:

- True label $y_i \in \{1, \ldots, g\}$, one-hot encoded as $\mathbf{d}^{(i)} = (0, \ldots, 1, \ldots, 0)$
- Model outputs probability vector $\boldsymbol{\pi}(\mathbf{x}_i \mid \theta) = (\pi_1, \ldots, \pi_g)$

The cross-entropy between one-hot $\mathbf{d}^{(i)}$ and model $\boldsymbol{\pi}$:

$$H(\mathbf{d}^{(i)} \| \boldsymbol{\pi}) = -\sum_{k=1}^{g} d_k^{(i)} \log \pi_k(\mathbf{x}_i \mid \theta) = -\log \pi_{y_i}(\mathbf{x}_i \mid \theta)$$

Cross-Entropy Loss in Classification

Multi-class classification with g classes. For observation i:

- True label $y_i \in \{1, \ldots, g\}$, one-hot encoded as $\mathbf{d}^{(i)} = (0, \ldots, 1, \ldots, 0)$
- Model outputs probability vector $\boldsymbol{\pi}(\mathbf{x}_i \mid \theta) = (\pi_1, \ldots, \pi_g)$

The cross-entropy between one-hot $\mathbf{d}^{(i)}$ and model $\boldsymbol{\pi}$:

$$H(\mathbf{d}^{(i)} \| \boldsymbol{\pi}) = -\sum_{k=1}^{g} d_k^{(i)} \log \pi_k(\mathbf{x}_i \mid \theta) = -\log \pi_{y_i}(\mathbf{x}_i \mid \theta)$$

Summing over all observations:

$$\mathcal{R}(\theta) = -\sum_{i=1}^{n} \log \pi_{y_i}(\mathbf{x}_i \mid \theta) \qquad (= \text{negative log-likelihood} = \text{log-loss})$$

Cross-entropy loss IS log-loss IS negative log-likelihood.
Three names for the same thing. Softmax + cross-entropy = the standard recipe.

Binary Cross-Entropy = Bernoulli Log-Loss

Binary classification: $y \in \{0, 1\}$, model outputs $\pi(\mathbf{x}) = P(Y{=}1 \mid \mathbf{x})$.

$$L(y, \pi) = -y \log \pi(\mathbf{x}) - (1 - y) \log(1 - \pi(\mathbf{x}))$$

This is the cross-entropy between two Bernoulli distributions: $p = \text{Bern}(y)$ and $q = \text{Bern}(\pi(\mathbf{x}))$.

Connection to Lecture 5 (MLE): Logistic regression maximizes likelihood $\Leftrightarrow$ minimizes binary cross-entropy $\Leftrightarrow$ minimizes KL to the true conditional.

The cross-entropy loss in logistic regression **is** information-theoretic!

Entropy as Baseline Risk

What's the best a "dumb" model can do? Use class frequencies: $\pi_k = n_k/n$.
The average cross-entropy of this constant model:

$$\mathcal{R} = -\sum_{k=1}^{g} \frac{n_k}{n} \log \frac{n_k}{n} = -\sum_k \pi_k \log \pi_k = H(\boldsymbol{\pi})$$

Entropy of the class distribution = risk of the best constant model.

Any good classifier must beat this baseline. The **reduction** from $H(\boldsymbol{\pi})$ is what your model actually learns.

Class balance	Baseline entropy	Interpretation
$(0.5, 0.5)$	1 bit	Hard (max uncertainty)
$(0.9, 0.1)$	0.47 bits	Easy (already pretty sure)
$(0.99, 0.01)$	0.08 bits	Trivial (almost deterministic)

Forward vs Reverse KL

KL is not symmetric. The direction matters — a lot.

Two Directions, Two Behaviors

We want to approximate a complex distribution p with a simpler model q_ϕ.

Forward KL: $D_{\mathrm{KL}}(p\|q_\phi)$

- Averages $\log \frac{p}{q_\phi}$ under p
- Penalizes $q_\phi \approx 0$ where $p > 0$
- q_ϕ must **cover all** of p's mass
- Behavior: **mass-covering**
- Used in: supervised learning, MLE

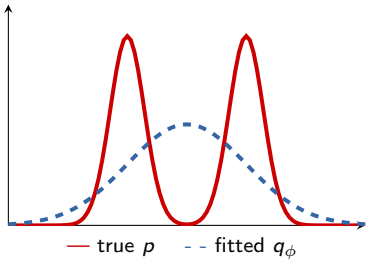
Reverse KL: $D_{\mathrm{KL}}(q_\phi\|p)$

- Averages $\log \frac{q_\phi}{p}$ under q_ϕ
- Penalizes $q_\phi > 0$ where $p \approx 0$
- q_ϕ avoids regions where p is small
- Behavior: **mode-seeking**
- Used in: variational inference

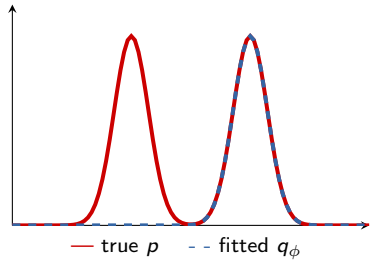
Forward KL: "I'd rather be too wide than miss anything."
Reverse KL: "I'd rather be too narrow than put mass where p doesn't."

Forward vs Reverse KL: Fitting a Gaussian to a Bimodal

Forward KL: mass-covering



Reverse KL: mode-seeking



Forward KL produces a **wide** fit covering both modes (but too flat).
Reverse KL locks onto **one mode** (sharp but misses the other).

Why Each Direction Is Used

	Forward KL: $D_{\mathrm{KL}}(p\|q_\phi)$	**Reverse KL:** $D_{\mathrm{KL}}(q_\phi\|p)$
Gradient	Requires samples from p	Requires samples from q_ϕ
Works when	p is the empirical distribution	p is unnormalized (up to Z)
Use case	Supervised learning (MLE)	Variational inference (VI)
Behavior	Mass-covering (inclusive)	Mode-seeking (exclusive)

Forward KL for supervised learning: we have data from p, so we can compute the gradient of $D_{\mathrm{KL}}(p\|q_\theta)$ from samples.

Reverse KL for variational inference: the posterior $p(\theta \mid \mathbf{X})$ is known only up to a constant, but $\nabla_\phi D_{\mathrm{KL}}(q_\phi\|p)$ doesn't need the normalizer.

Maximum Entropy Principle

Given partial knowledge, choose the distribution
that is **maximally uncertain** about everything else.

The Maximum Entropy Principle (Jaynes, 2003)

Suppose we know some constraints (e.g., $\mathbb{E}[g_m(X)] = \alpha_m$ for $m = 1, \ldots, M$). Many distributions satisfy these constraints. Which one should we pick?

MaxEnt Principle: Among all distributions satisfying the constraints, choose the one with **maximum entropy**.

Rationale: don't assume more than you know. (Occam's razor for distributions.)

The Maximum Entropy Principle (Jaynes, 2003)

Suppose we know some constraints (e.g., $\mathbb{E}[g_m(X)] = \alpha_m$ for $m = 1, \ldots, M$). Many distributions satisfy these constraints. Which one should we pick?

MaxEnt Principle: Among all distributions satisfying the constraints, choose the one with **maximum entropy**.

Rationale: don't assume more than you know. (Occam's razor for distributions.)

Solution (Lagrangian):

$$p^*(x) = \frac{1}{Z} \exp\left(\sum_{m=1}^{M} \lambda_m \, g_m(x)\right) \qquad Z = \sum_x \exp\left(\sum_m \lambda_m \, g_m(x)\right)$$

The MaxEnt distribution is always an **exponential family** (Gibbs distribution)!
This is no coincidence — exponential families **are** the MaxEnt distributions.

MaxEnt Recovers Familiar Distributions

Different constraints $\Rightarrow$ different MaxEnt distributions:

Constraint(s)	**MaxEnt distribution**	**Connection**
None (fixed support $\{1, \ldots, g\}$)	Uniform	Max entropy $= \log g$
$\mathbb{E}[X] = \mu$ (on $\{1, \ldots, g\}$)	Exponential/Gibbs	Biased die
$\mathbb{E}[X] = \mu$, $\mathrm{Var}(X) = \sigma^2$	**Gaussian** $N(\mu, \sigma^2)$	Most common in ML!
$\mathbb{E}[X] = 1/\lambda$ (on $[0, \infty)$)	**Exponential**(λ)	Memoryless

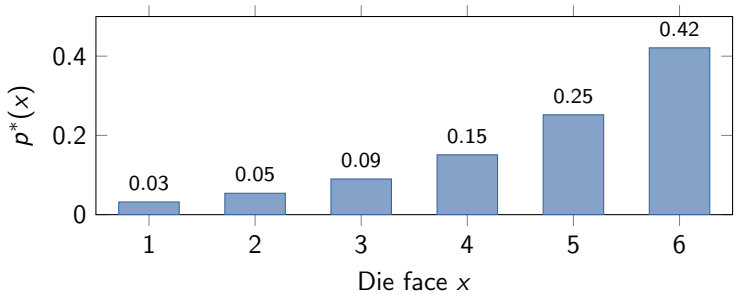
Why the Gaussian is everywhere in ML: If you only know the mean and variance, the Gaussian is the **least informative** distribution compatible with that knowledge. Any other choice would smuggle in extra assumptions.

MaxEnt Example: The Biased Die

A 6-sided die with $\mathbb{E}[X] = 4.8$ (heavier than the usual 3.5). What's the MaxEnt distribution?

$$p^*(x) = \frac{e^{\lambda x}}{Z(\lambda)}, \qquad Z(\lambda) = \sum_{x=1}^{6} e^{\lambda x}, \qquad \sum_{x=1}^{6} x\, p^*(x) = 4.8$$

Solving numerically: $\lambda \approx 0.514$.



Exponential tilt toward higher faces — the "least opinionated" way to have $\mathbb{E}[X] = 4.8$.

Mutual Information

How much does knowing X tell us about Y?

A universal measure of dependence.

Joint and Conditional Entropy

Joint entropy: Total uncertainty in (X, Y) together:

$$H(X, Y) = -\sum_{x,y} p(x, y) \log p(x, y)$$

Conditional entropy: Remaining uncertainty in Y after observing X:

$$H(Y \mid X) = -\sum_{x,y} p(x, y) \log p(y \mid x) = \mathbb{E}_X[H(Y \mid X{=}x)]$$

Chain Rule: $H(X, Y) = H(X) + H(Y \mid X)$

Total uncertainty = uncertainty in X + remaining uncertainty in Y given X.

Generalizes: $H(X_1, \ldots, X_n) = \sum_{i=1}^{n} H(X_i \mid X_1, \ldots, X_{i-1})$.

Proof: $\log p(x, y) = \log p(x) + \log p(y|x)$. Take $-\mathbb{E}$ of both sides.

Mutual Information: Definition

Question: How much does knowing X **reduce** our uncertainty about Y?

Mutual Information:

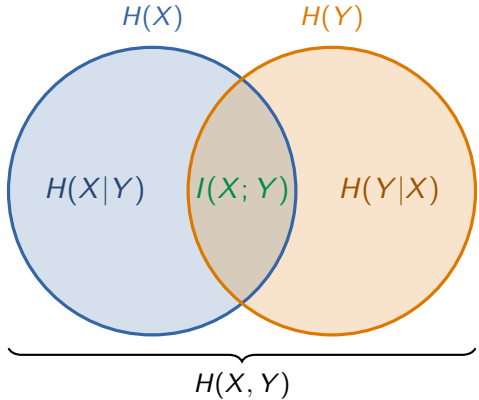
$$I(X;Y) \quad = \quad H(X) - H(X \mid Y) \quad = \quad H(Y) - H(Y \mid X)$$

$$= \quad \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)} \quad = \quad D_{\mathrm{KL}}\big(p(x,y) \;\|\; p(x)\,p(y)\big)$$

"KL divergence between the joint and the product of marginals."

MI measures how much the joint distribution **deviates from independence**.
$I(X;Y) = 0$ iff $X \perp\!\!\!\perp Y$. The larger $I(X;Y)$, the more they "know" about each other.

The Information Diagram



$I(X;Y) = H(X) + H(Y) - H(X,Y)$ — the overlap = shared information.

Properties of Mutual Information

1. Non-negative: $I(X;Y) \geq 0$ (since $D_{\mathrm{KL}} \geq 0$)

2. Zero iff independent: $I(X;Y) = 0 \Leftrightarrow X \perp\!\!\!\perp Y$

3. Symmetric: $I(X;Y) = I(Y;X)$ (unlike KL!)

4. Self-information: $I(X;X) = H(X)$

5. Bounded: $I(X;Y) \leq \min\{H(X), H(Y)\}$ (for discrete RVs)

6. Invariant: under invertible smooth transformations of X or Y

Conditioning reduces entropy ("information can't hurt"):

$H(X \mid Y) \leq H(X)$, with equality iff $X \perp\!\!\!\perp Y$.

Proof: $0 \leq I(X;Y) = H(X) - H(X \mid Y)$.

Worked Example: Computing MI

Joint distribution $p(x, y)$:

$X \backslash Y$	y_1	y_2	y_3	y_4
x_1	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{32}$
x_2	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{32}$
x_3	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$
x_4	$\frac{1}{4}$	0	0	0

Marginals:

- $p_X = (1/4,\ 1/4,\ 1/4,\ 1/4)$
- $p_Y = (1/2,\ 1/4,\ 1/8,\ 1/8)$

Entropies:

- $H(X) = 2$ bits (uniform over 4)
- $H(Y) = 7/4$ bits
- $H(X, Y) = 27/8$ bits

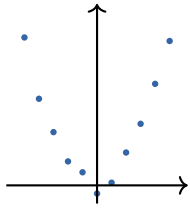
Mutual information:

$$I(X; Y) = H(X) + H(Y) - H(X, Y)$$

$$= 2 + \tfrac{7}{4} - \tfrac{27}{8} = \boxed{\tfrac{3}{8} \text{ bits}}$$

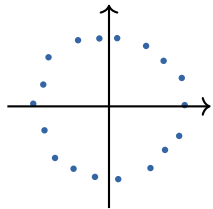
MI vs Correlation: Why MI Is More General

Pearson correlation only measures **linear** dependence. MI captures **any** dependence.



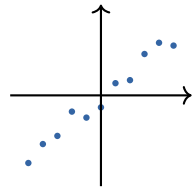
$Y = X^2 + \varepsilon$

$\rho \approx 0,\ I > 0$



Circle

$\rho \approx 0,\ I > 0$



Linear

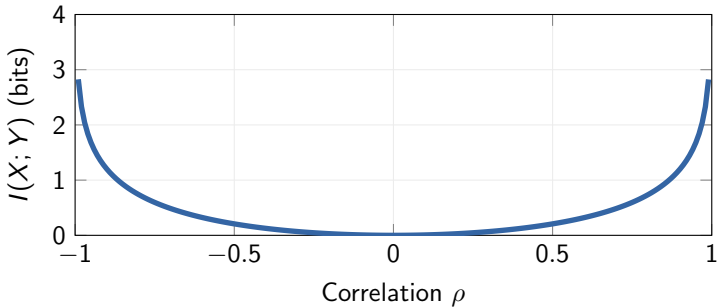
$\rho > 0,\ I > 0$

Correlation = 0 does NOT mean independence. MI detects **any** dependence — linear, nonlinear, or otherwise.

MI for Correlated Gaussians

For $(X, Y) \sim N\left(\mathbf{0}, \begin{pmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{pmatrix}\right)$:

$$I(X;Y) = -\frac{1}{2}\log_2(1-\rho^2)$$

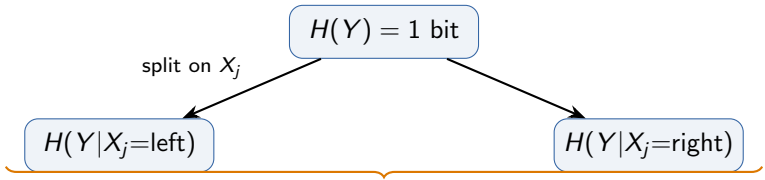


$\rho = 0 \Rightarrow I = 0$ (Gaussian: uncorrelated = independent).
$|\rho| \to 1 \Rightarrow I \to \infty$ (perfect linear dependence).

MI in ML: Information Gain in Decision Trees

At each node of a decision tree, we pick the feature that **maximizes information gain**:

$$\mathrm{IG}(Y; X_j) = H(Y) - H(Y \mid X_j) = I(Y; X_j)$$



$$\textbf{IG} = H(Y) - H(Y|X_j) = I(Y; X_j)$$

Split on the feature with highest MI with the target.
Exactly the criterion in ID3, C4.5, CART (entropy impurity). Feature selection: rank features by $I(X_j; Y)$, pick top ones.

The Information Theory Toolbox for ML

Concept	Formula	ML Role
Entropy $H(p)$	$-\sum p \log p$	Baseline risk, impurity
Cross-entropy $H(p\|q)$	$-\sum p \log q$	**Loss function** (log-loss)
KL divergence D_{KL}	$\sum p \log \frac{p}{q}$	MLE, variational inference
Mutual info $I(X;Y)$	$D_{\mathrm{KL}}(p_{xy} \| p_x p_y)$	Feature selection, info gain
Forward KL	$D_{\mathrm{KL}}(p \| q_\theta)$	Supervised learning
Reverse KL	$D_{\mathrm{KL}}(q_\phi \| p)$	Variational inference
MaxEnt	$\arg\max H$ s.t. constraints	Gaussian, exp family

Key takeaway: Information theory is not just "theory" — it directly gives us the loss functions, optimization objectives, and model selection criteria we use every day in ML.

Homework

1. Show that minimizing cross-entropy $H(\hat{p}\|q_\theta)$ over training data is equivalent to maximizing the log-likelihood.
2. For the joint distribution in the worked example, verify $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = 3/8$ bits.
3. Binary classification with $P(Y{=}1) = 0.7$.
   (a) Compute $H(Y)$.
   (b) If $H(Y|X) = 0.5$ bits after observing feature X, what is $I(X;Y)$?
   (c) Compare: feature X' with $I(X';Y) = 0.01$ bits. Which is more useful?
4. Explain intuitively why forward KL is "mass-covering" and reverse KL is "mode-seeking."

Questions?