

ECE 590.17: Lecture 3 – Probability and Information Measures

Graphical Models and Inference for Machine Learning

Duke University, Spring 2026

History: Written by Henry Pfister (2026).

Last Modified: 01/28/2026

Outline of lecture:

3.1	Random variables and distributions	1
3.2	Entropy	2
3.3	The typical set and compression	3
3.4	Cross-entropy and Kullback-Leibler (KL) divergence	4
3.5	Mutual information	5
3.6	Log-likelihood and inference	6
3.7	Summary of key identities	6
3.8	Optional: Entropy and data compression	6
3.9	Deferred Proofs	8
	3.9.1 Fundamental Inequality	8
3.10	Convexity and Concavity	9
	3.10.1 Convexity of KL Divergence	9

3.1 Random variables and distributions

We work primarily with discrete random variables. A discrete random variable X is specified by a finite alphabet \mathcal{X} and a probability mass function $p_X(x) := \mathbb{P}(X = x)$ with $\sum_{x \in \mathcal{X}} p_X(x) = 1$.

Joint and conditional distributions. For (X, Y) taking values in $\mathcal{X} \times \mathcal{Y}$, the joint distribution is $p_{X,Y}(x, y)$, the X -marginal is $p_X(x) = \sum_y p_{X,Y}(x, y)$, and the conditional distribution of Y given X is $p_{Y|X}(y|x) = p_{X,Y}(x, y)/p_X(x)$, whenever $p_X(x) > 0$.

Expectation and variance. For any function $f : \mathcal{X} \rightarrow \mathbb{R}$, we define

$$\mathbb{E}[f(X)] := \sum_{x \in \mathcal{X}} p_X(x) f(x), \quad \text{Var}(f(X)) := \mathbb{E}[(f(X) - \mathbb{E}[f(X)])^2] = \mathbb{E}[f(X)^2] - \mathbb{E}[f(X)]^2.$$

Expectation is linear: $\mathbb{E}[af(X) + bg(X)] = a\mathbb{E}[f(X)] + b\mathbb{E}[g(X)]$.

Independence. Random variables X and Y are independent iff $p_{X,Y}(x, y) = p_X(x)p_Y(y)$ for all x, y . Equivalently, $p_{Y|X}(y|x) = p_Y(y)$.

Notation. For distributions p_X and $p_{X,Y}$, the subscripts may be dropped when they can be inferred from the context.

3.2 Entropy

The Shannon entropy (in bits) of X is

$$H(X) = \sum_{x \in \mathcal{X}} p_X(x) \log_2 \frac{1}{p_X(x)} = - \sum_{x \in \mathcal{X}} p_X(x) \log_2 p_X(x).$$

Entropy measures uncertainty and be interpreted operationally as the fundamental lower bound for compression: An i.i.d. sequence X_1, X_2, \dots, X_n can be compressed to $nH(X) + o(n)$ bits but not any fewer.

Basic properties.

- $H(X) \geq 0$, with equality iff X is constant.
- Concavity: $H(p)$ is concave in the distribution p .
- $H(X) \leq \log_2 |\mathcal{X}|$, with equality for uniform X .

Proof sketches.

- **Nonnegativity:** Each term $-p(x) \log_2 p(x) \geq 0$ because $0 \leq p(x) \leq 1$ implies $\log_2 p(x) \leq 0$. Equality holds iff one probability is 1 and the rest are 0.
- **Concavity:** The function $x \mapsto -x \log_2 x$ is concave on $[0, 1]$. Therefore, for $p_\lambda = \lambda p + (1-\lambda)q$, $H(p_\lambda) = \sum_x -p_\lambda(x) \log_2 p_\lambda(x)$ satisfies $H(p_\lambda) \geq \lambda H(p) + (1-\lambda)H(q)$.
- **Maximum at uniform:** Using the fundamental inequality $\log_2(x) \leq \log_2(e)(x-1)$, we have

$$\begin{aligned} \sum_x p(x) \log_2 \frac{1}{p(x)} &= \sum_x p(x) \log_2 \left(\frac{|\mathcal{X}|}{p(x)|\mathcal{X}|} \right) \\ &= \log_2(|\mathcal{X}|) + \sum_x p(x) \log_2 \left(\frac{1}{p(x)|\mathcal{X}|} \right) \\ &\leq \log_2(|\mathcal{X}|) + \sum_x p(x) \log_2(e) \left(\frac{1}{p(x)|\mathcal{X}|} - 1 \right) \quad \text{Fundamental Inq.} \\ &= \log_2(|\mathcal{X}|) + \log_2(e) - \log_2(e) \end{aligned}$$

Joint and conditional entropy. For (X, Y) , we have

$$H(X, Y) = - \sum_{x,y} p_{X,Y}(x, y) \log_2 p_{X,Y}(x, y),$$

and the conditional entropy is

$$H(Y|X) = \sum_x p_X(x) H(p_{Y|X}(\cdot|x)).$$

Chain rule.

$$\begin{aligned}
 H(X, Y) &= - \sum_{x, y} p_{X, Y}(x, y) \log_2 p_{X, Y}(x, y) \\
 &= - \sum_{x, y} p_{X, Y}(x, y) \log_2 (p_X(x) p_{Y|X}(y|x)) \\
 &= - \sum_x p_X(x) \log_2 p_X(x) - \sum_{x, y} p_{X, Y}(x, y) \log_2 p_{Y|X}(y|x) \\
 &= H(X) + H(Y|X).
 \end{aligned}$$

3.3 The typical set and compression

Typical set. Let $X^n = (X_1, \dots, X_n)$ be i.i.d. with distribution p_X . Then, we have

$$p(x^n) \equiv p_{X^n}(x^n) = \prod_{i=1}^n p_X(x_i).$$

For $\epsilon > 0$, define the ϵ -typical set

$$\mathcal{T}_\epsilon^{(n)} = \left\{ x^n \in \mathcal{X}^n : 2^{-n(H(X)+\epsilon)} \leq p(x^n) \leq 2^{-n(H(X)-\epsilon)} \right\}.$$

We will see that the typical set contains almost all the probability and the entropy $H(X)$ is the exponential growth rate (in n) of its cardinality, which connects entropy to counting.

Theorem 1 (Asymptotic Equipartition Property (AEP)). *The typical set satisfies:*

- For all $x^n \in \mathcal{T}_\epsilon^{(n)}$, we have $2^{-n(H(X)+\epsilon)} \leq p(x^n) \leq 2^{-n(H(X)-\epsilon)}$.
- For any $\epsilon > 0$, we have $\mathbb{P}(X^n \in \mathcal{T}_\epsilon^{(n)}) \rightarrow 1$ as $n \rightarrow \infty$.
- We have $(1 - o(1))2^{n(H(X)-\epsilon)} \leq |\mathcal{T}_\epsilon^{(n)}| \leq 2^{n(H(X)+\epsilon)}$.

Proof. The first item follows directly from the definition of $\mathcal{T}_\epsilon^{(n)}$. For the second item, define $Z_i = -\log_2 p_X(X_i)$, so that

$$-\frac{1}{n} \log_2 p_{X^n}(X^n) = \frac{1}{n} \sum_{i=1}^n Z_i.$$

Since the X_i are i.i.d., the Z_i are i.i.d. with $\mathbb{E}[Z_i] = H(X)$. By the weak law of large numbers, for any $\epsilon > 0$, we have

$$\mathbb{P}\left(\left| -\frac{1}{n} \log_2 p_{X^n}(X^n) - H(X) \right| > \epsilon\right) \rightarrow 0,$$

which is equivalent to $\mathbb{P}(X^n \in \mathcal{T}_\epsilon^{(n)}) \rightarrow 1$. For the cardinality bounds, note that

$$1 \geq \mathbb{P}(X^n \in \mathcal{T}_\epsilon^{(n)}) = \sum_{x^n \in \mathcal{T}_\epsilon^{(n)}} p(x^n) \geq |\mathcal{T}_\epsilon^{(n)}| \cdot 2^{-n(H(X)+\epsilon)},$$

so $|\mathcal{T}_\epsilon^{(n)}| \leq 2^{n(H(X)+\epsilon)}$. Also,

$$\mathbb{P}(X^n \in \mathcal{T}_\epsilon^{(n)}) \leq \sum_{x^n \in \mathcal{T}_\epsilon^{(n)}} 2^{-n(H(X)-\epsilon)} = |\mathcal{T}_\epsilon^{(n)}| \cdot 2^{-n(H(X)-\epsilon)},$$

so $|\mathcal{T}_\epsilon^{(n)}| \geq \mathbb{P}(X^n \in \mathcal{T}_\epsilon^{(n)}) \cdot 2^{n(H(X)-\epsilon)}$ and $\mathbb{P}(X^n \in \mathcal{T}_\epsilon^{(n)}) = 1 - o(1)$ gives the lower bound. \square

Compression limit intuition. The AEP says that, with high probability, X^n lies in a set of size about $2^{nH(X)}$, and all typical sequences have roughly equal probability. Thus we can assign codewords only to the typical set and use about $\log_2 |\mathcal{T}_\epsilon^{(n)}| \approx nH(X)$ bits to describe them, with vanishing error probability. Conversely, any lossless code that succeeds with high probability must distinguish roughly $2^{nH(X)}$ typical sequences, so it needs at least $nH(X)$ bits per block asymptotically. This pins the compression limit at $H(X)$ bits per symbol.

Entropy rate. For a stochastic process $\{X_t\}$, the entropy rate is

$$h_X = \lim_{n \rightarrow \infty} \frac{1}{n} H(X_1, \dots, X_n),$$

when the limit exists. The process is called *information stable* if the empirical entropy

$$\frac{1}{n} \log \frac{1}{p_{X^n}(X^n)} \rightarrow h_X \quad \text{in probability.}$$

An ergodic Markov chain, with transition probabilities $p_{X_{n+1}|X_n}(x'|x)$ and stationary distribution $\pi(x)$, is information stable with

$$h_X = - \sum_x \pi(x) \sum_{x'} p(x'|x) \log_2 p(x'|x).$$

3.4 Cross-entropy and Kullback-Leibler (KL) divergence

Given two distributions p and q over the same alphabet, the cross-entropy of q relative to p is

$$H(p, q) := - \sum_x p(x) \log_2 q(x).$$

We will see that it is minimized over q by choosing $q = p$ and, in that case, it equals the entropy $H(p)$. Here, we are overloading notation and using $H(p)$ to denote the entropy of $X \sim p$ and $H(p, q)$ to denote the cross entropy in terms of the distributions instead of related random variables.

The KL divergence is defined to be

$$D(p||q) = \sum_x p(x) \log_2 \frac{p(x)}{q(x)} = H(p, q) - H(p).$$

Nonnegativity. The divergence is nonnegative and equals zero iff $p(x) = q(x)$ for all $x \in \mathcal{X}$.

$$\begin{aligned} D(p||q) &= - \sum_x p(x) \log_2 \frac{q(x)}{p(x)} \\ &\geq - \sum_x p(x) \log_2(e) \left(\frac{q(x)}{p(x)} - 1 \right) && \text{Fundamental Inq.} \\ &= - \log_2(e) \left(\sum_x p(x) - q(x) \right) = 0. \end{aligned}$$

For each x , equality occurs iff $p(x) = q(x)$ or $p(x) = 0$. Thus, $D(p||q) = 0$ iff $p(x) = q(x)$ for $x \in \mathcal{X}$.

Operational meanings:

- **Compression.** An optimal compression code matched to q assigns the sequence x^n a code-word that is $-\log_2 q(x^n) + o(n)$ bits long. Thus, the AEP implies that the expected code length on data generated by q approaches $H(q)$ bits/symbol. When the string is drawn iid from p , the expected code length (in bits/symbol) approaches the cross-entropy

$$H(p, q) = \mathbb{E}_p[-\log_2 q(X)] = - \sum_{x \in \mathcal{X}} p(x) \log_2 q(x).$$

Since we have

$$H(p, q) - H(p) = D(p||q),$$

it follows that $D(p||q)$ quantifies the penalty for model mismatch. It is minimized at $q = p$, reflecting that model mismatch only increases average description length.

- **Hypothesis testing.** Given n i.i.d. samples either from p (the null hypothesis) or q (the alternate hypothesis), the optimal error probability decays exponentially with n . If the type-I error (falsely rejecting the null) goes to zero, then Stein's lemma states that the best exponential decay rate of type-II error (falsely choosing the null) is $D(p||q)$. Thus, for a test that usually identifies the null hypothesis correctly, the probability of falsely rejecting the alternate hypothesis (i.e., choosing p when q is true) decays like $2^{-nD(p||q)}$.

Large deviations interpretation. Consider Bernoulli models p, q where $p \sim \text{Bern}(r)$ and $q \sim \text{Bern}(s)$. Under q , the probability of observing an empirical frequency close to r decays like $2^{-nD(p||q)}$. More generally, KL divergence acts as a rate function for rare events and model misspecification.

For Bernoulli random variables, we define

$$\begin{aligned} h(r) &:= H(\text{Bern}(r)) = -r \log_2(r) - (1-r) \log_2(1-r) \\ d(r||s) &:= D(\text{Bern}(r)||\text{Bern}(s)) = -r \log_2 \frac{r}{s} - (1-r) \log_2 \frac{1-r}{1-s}. \end{aligned}$$

Proof sketch. For $k = rn$ successes under parameter s , the probability is $\binom{n}{k} s^k (1-s)^{n-k}$. Stirling's approximation gives $\binom{n}{rn} \approx 2^{nH(p)} = 2^{nh(r)}$, so

$$\begin{aligned} \mathbb{P}_q(k = rn) &\approx 2^{nh(r)} s^{rn} (1-s)^{(1-r)n} \\ &= 2^{-n(r \log_2(r) + (1-r) \log_2(1-r) - r \log_2(s) - (1-r) \log_2(1-s))} \\ &= 2^{-nd(r||s)} = 2^{-nD(\text{Bern}(r)||\text{Bern}(s))}. \end{aligned}$$

3.5 Mutual information

The mutual information $I(X; Y)$ between X and Y measures the dependence between these random variables and is defined by

$$I(X; Y) = D(p_{X,Y} || p_X p_Y) = H(X) + H(Y) - H(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X).$$

It is nonnegative and equals zero iff X and Y are independent.

Conditional mutual information. For X, Y, Z ,

$$I(X; Y|Z) = H(X|Z) - H(X|Y, Z).$$

This will be important for conditional independence structure in factor graphs.

3.6 Log-likelihood and inference

For data x and model q_θ , the log-likelihood is $\log q_\theta(x)$. Maximum likelihood estimation chooses θ to maximize average log-likelihood, which is equivalent to minimizing cross-entropy between the empirical distribution and q_θ .

KL and model fitting. If the true distribution is p , then

$$\mathbb{E}_p[-\log q_\theta(X)] = H(p, q_\theta) = H(p) + D(p||q_\theta).$$

This equals the number of bits/symbol we need to compress X when it is drawn from p but q_θ is used to compress it. The maximum likelihood rule minimizes $D(p||q_\theta)$ over the model family and thus minimizes the number of bits required to compress the observations given the model. This is related to the minimum description length (MDL) principle which seeks the model θ with the minimum length description when the cost of communicating θ itself is included.

3.7 Summary of key identities

$$\begin{aligned} H(X, Y) &= H(X) + H(Y|X), \\ I(X; Y) &= H(X) - H(X|Y) = D(p_{X,Y}||p_X p_Y), \\ D(p||q) &= H(p, q) - H(p) \geq 0. \end{aligned}$$

3.8 Optional: Entropy and data compression

Consider a source that generates a sequence of symbols taking values in a finite alphabet \mathcal{X} . Such a source is typically modeled by a random process $\{X_t\}_{t \in \mathbb{N}}$. For the purpose of communication and storage, it is desirable to encode this sequence using as few bits as possible. If exact reconstruction is required, then this is known as **lossless source coding**.

The most natural approach to this problem is to encode length- N source blocks (X_1, \dots, X_N) into variable-length blocks of bits. Let $\{0, 1\}^* = \cup_{n=0}^{\infty} \{0, 1\}^n$ denote the set of finite-length binary strings. Then, the source encoder is a function

$$\begin{aligned} w: \mathcal{X}^N &\rightarrow \{0, 1\}^* \\ \underline{x} &\mapsto w(\underline{x}). \end{aligned}$$

If the source sequence consists of the length- N blocks $\underline{x}^{(1)}, \underline{x}^{(2)}, \dots, \underline{x}^{(r)}$, then the encoded sequence is the concatenation $w(\underline{x}^{(1)})w(\underline{x}^{(2)}) \dots w(\underline{x}^{(r)})$. Since the output sequence does not add markers between blocks, one must choose the w carefully to guarantee decodability. The standard approach is use to **prefix-free** (or **instantaneous**) codes where no codeword is a prefix of another codeword. This allows the decoder to uniquely reconstruct the codeword boundaries.

An important property of a code is its average length. If $l_w(\underline{x})$ is the length of $w(\underline{x})$ in bits, then the average length of an encoded block is

$$L(w) = \sum_{\underline{x} \in \mathcal{X}^N} P_N(\underline{x}) l_w(\underline{x}) \quad \text{bits,}$$

where $P_N(\underline{x})$ is the distribution of length- N blocks drawn iid from the source.

Any prefix-free code can be represented by a binary tree whose leaf nodes are labeled by codewords. To construct a prefix-free code, one can draw a binary tree and sequentially assign \underline{x} values to nodes. After each assignment, all children of the assigned node are removed.

Exercise 1. *Is there a prefix-free code with codeword lengths 1, 2, 3, 3? How about 2, 2, 3, 3, 3, 4, 4, 4? Try constructing a code for each case.*

Lemma 2 (Kraft Inequality). *A prefix-free source code with length function $l_w(\underline{x})$ exists iff*

$$\sum_{\underline{x} \in \mathcal{X}^N} 2^{-l_w(\underline{x})} \leq 1.$$

Proof. Let $l_{\max} = \max_{\underline{x} \in \mathcal{X}^N} l_w(\underline{x})$ and recall that a binary tree has exactly 2^l nodes at depth- l . To construct a code, one starts with the complete binary tree of depth l_{\max} . Then, for each $\underline{x} \in \text{supp}(P_N)$ (in order of increasing length), one assigns \underline{x} to a codeword $w(\underline{x})$ of length $l_w(\underline{x})$. For an \underline{x} with length $l_w(\underline{x})$, one finds an available node at depth $l_w(\underline{x})$, assigns the binary label of that node to $w(\underline{x})$, and then removes all children of that node. Assigning a codeword of length $l_w(\underline{x})$ removes exactly $2^{l-l_w(\underline{x})}$ nodes at depth- l for $l \geq l_w(\underline{x})$. Thus, this process succeeds up to depth- l if and only if

$$\sum_{\underline{x}: l_w(\underline{x}) \leq l} 2^{l-l_w(\underline{x})} \leq 2^l.$$

Dividing by 2^l , one sees that this condition is most restrictive for $l = l_{\max}$. Choosing $l = l_{\max}$ and dividing by $2^{l_{\max}}$ gives the desired result. \square

Theorem 3 (Source Coding Theorem). *For the distribution $P_N(\underline{x})$, let L_N^* be average length of an encoded block for the prefix-free code with the minimum average length. Then,*

$$H(\underline{X}) \leq L_N^* \leq H(\underline{X}) + 1.$$

Proof. Let $l_w(\underline{x})$ be the length function for a valid prefix-free code and define $Q_N(\underline{x}) = 2^{-l_w(\underline{x})}$. Since $l_w(\underline{x})$ must satisfy the Kraft inequality, it follows that $\sum_{\underline{x}} Q_N(\underline{x}) \leq 1$. Since the divergence is non-negative (which holds even if $\sum_{\underline{x}} q(\underline{x}) \leq 1$), we see that the average code length, L , satisfies

$$\begin{aligned} L - H(P_N) &= \sum_{\underline{x} \in \mathcal{X}} P_N(\underline{x}) \left(l_w(\underline{x}) - \log_2 \frac{1}{P_N(\underline{x})} \right) \\ &= \sum_{\underline{x} \in \mathcal{X}} P_N(\underline{x}) \left(\log_2 \frac{P_N(\underline{x})}{Q_N(\underline{x})} \right) \\ &= D(P_N \| Q_N) \\ &\geq 0. \end{aligned} \tag{3.1}$$

If we choose $l_w(\underline{x})$ to be the length function for a code that achieves the optimal $L = L_N^*$, then this implies that $L_N^* \geq H(P_N)$. To achieve the upper bound, we design a code with $l_w(\underline{x}) =$

$\lceil -\log_2 P_N(\underline{x}) \rceil$. This choice of $l_w(\underline{x})$ satisfies the Kraft inequality because $2^{-l_w(\underline{x})} \leq P_N(\underline{x})$ and the sum over all \underline{x} is upper bounded by 1. Computing the upper bound on L_N^* gives

$$\begin{aligned} L_N^* &\leq \sum_{\underline{x} \in \mathcal{X}} P_N(\underline{x}) \lceil -\log_2 P_N(\underline{x}) \rceil \\ &\leq \sum_{\underline{x} \in \mathcal{X}} P_N(\underline{x}) \left(\log_2 \frac{1}{P_N(\underline{x})} + 1 \right) \\ &= H(\underline{X}) + 1. \end{aligned}$$

Together, these complete the proof. \square

Remark 4. This shows that one operational definition of the entropy is “the minimum average length of any variable length code that can be used to reconstruct \underline{X} ”.

Remark 5. In theory, the source coding theorem proves one can achieve optimal compression rate of $H(X)$ for i.i.d. sequences. To see this, we observe that $H(P_N) = NH(X_1)$. Thus, by increasing N , the constructive upper bound in the theorem gives a compression rate (i.e., bits per source symbol) of

$$\frac{L_N^*}{N} \leq H(X_1) + \frac{1}{N}.$$

Example 6. Let us consider what happens if a code is designed for a different distribution, Q_N , and then used with the distribution P_N . Since $L - H(P_N) \geq D(P_N \| Q_N)$, we see that the average code length, L , must satisfy $L \geq H(P_N) + D(P_N \| Q_N)$. On the other hand, if the lengths are chosen to be $l_w(\underline{x}) = \lceil -\log_2 Q_N(\underline{x}) \rceil$, then the average length satisfies

$$\begin{aligned} L &= \sum_{\underline{x} \in \mathcal{X}} P_N(\underline{x}) \log_2 \lceil -\log_2 Q_N(\underline{x}) \rceil \\ &\leq \sum_{\underline{x} \in \mathcal{X}} P_N(\underline{x}) \left(\log_2 \frac{1}{Q_N(\underline{x})} + 1 \right) \\ &= H(\underline{X}) + 1 + \sum_{\underline{x} \in \mathcal{X}} P_N(\underline{x}) \left(\log_2 \frac{P_N(\underline{x})}{Q_N(\underline{x})} \right) \\ &= H(\underline{X}) + 1 + D(P_N \| Q_N). \end{aligned}$$

Remark 7. This shows that one operational definition of the divergence $D(P_N \| Q_N)$ is “the increase in average length associated with designing a code for Q_N when the true distribution is P_N ”.

3.9 Deferred Proofs

3.9.1 Fundamental Inequality

For any base $b > 0$ and $x > 0$,

$$\left(1 - \frac{1}{x} \right) \log_b(e) \leq \log_b(x) \leq (x - 1) \log_b(e)$$

with equalities on both sides if, and only if, $x = 1$. For the natural log, this simplifies to

$$\left(1 - \frac{1}{x} \right) \leq \ln(x) \leq (x - 1)$$

Proof of upper bound:

$$\begin{aligned}
 x \in (1, \infty) &\implies (x-1) - \ln(x) = \int_1^x \underbrace{\left(1 - \frac{1}{u}\right)}_{\text{strictly positive}} du > 0 \\
 x \in (0, 1) &\implies (x-1) - \ln(x) = \int_x^1 \underbrace{\left(\frac{1}{u} - 1\right)}_{\text{strictly positive}} du > 0
 \end{aligned}$$

Proof of lower bound: Let $y = 1/x$ and apply upper bound

$$\ln(y) \leq y - 1 \iff 1 - y \leq \ln\left(\frac{1}{y}\right) \iff 1 - \frac{1}{x} \leq \ln(x)$$

3.10 Convexity and Concavity

Definition 8. A subset $A \subseteq \mathbb{R}^n$ is *convex* if, for all $x, y \in A$ and $\lambda \in [0, 1]$, we have $\lambda x + (1-\lambda)y \in S$.

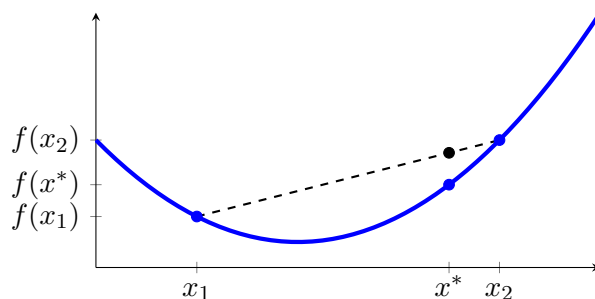
Definition 9. For a convex set $A \subseteq \mathbb{R}^d$, a function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ is *convex* on A if, for every $x_1, x_2 \in A$ and $0 \leq \lambda \leq 1$, we have

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2).$$

The function is *strictly convex* if equality holds only if $\lambda = 0$ or $\lambda = 1$. A function $f(x)$ is called (strictly) concave if $-f(x)$ is (strictly) convex.

Remark 10. It is easy to verify that any positive linear combination of convex functions is convex and strict convexity is guaranteed by including even one strictly convex function. If $f(x)$ is twice differentiable on A , then $f(x)$ is convex if and only if its Hessian matrix $\nabla^2 f(x)$ is positive semidefinite on A . For a real function with $A = (a, b)$, this is equivalent to $f''(x) \geq 0$ for all $x \in (a, b)$.

Illustration of convexity. Let $x^* = \lambda x_1 + (1-\lambda)x_2$



3.10.1 Convexity of KL Divergence

The divergence is convex in the pair (p, q) .

Log-Sum Inequality. Let $a_i, b_i \geq 0$ for $i = 1, \dots, n$ with $\sum_i a_i > 0$ and $\sum_i b_i > 0$. Then

$$\sum_{i=1}^n a_i \log_2 \frac{a_i}{b_i} \geq \left(\sum_{i=1}^n a_i \right) \log_2 \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i},$$

with equality iff a_i/b_i is constant for all i with $a_i b_i > 0$.

Proof. Let $A = \sum_i a_i$ and $B = \sum_i b_i$, and define $p_i = a_i/A$ and $q_i = b_i/B$. Then

$$D(p\|q) = \sum_i p_i \log_2 \frac{p_i}{q_i} \geq 0.$$

Multiplying by A gives

$$A \sum_i p_i \log_2 \frac{B a_i}{A b_i} = \sum_i a_i \log_2 \frac{a_i}{b_i} - A \log_2 \frac{A}{B} \geq 0,$$

which is the desired inequality. Equality holds iff $p = q$ on the support, i.e., a_i/b_i is constant. \square

Proof sketch. Let $\lambda \in [0, 1]$ and define $p_\lambda = \lambda p_1 + (1 - \lambda)p_2$ and $q_\lambda = \lambda q_1 + (1 - \lambda)q_2$. Apply the log-sum inequality to the two-term sums $a_1 = \lambda p_1(i)$, $a_2 = (1 - \lambda)p_2(i)$ and $b_1 = \lambda q_1(i)$, $b_2 = (1 - \lambda)q_2(i)$ to obtain

$$(\lambda p_1(i) + (1 - \lambda)p_2(i)) \log_2 \frac{\lambda p_1(i) + (1 - \lambda)p_2(i)}{\lambda q_1(i) + (1 - \lambda)q_2(i)} \leq \lambda p_1(i) \log_2 \frac{p_1(i)}{q_1(i)} + (1 - \lambda)p_2(i) \log_2 \frac{p_2(i)}{q_2(i)}.$$

Summing over i yields

$$D(p_\lambda\|q_\lambda) \leq \lambda D(p_1\|q_1) + (1 - \lambda)D(p_2\|q_2),$$

so $D(\cdot\|\cdot)$ is jointly convex in (p, q) .