Algorithm validation by information theory

Joachim Buhmann and Carlos Cotrini
Department of Computer Science
ETH Zürich

October 2, 2020

1 Introduction

These notes present an information-theoretic validation method called *posterior agreement*, proposed by Buhmann [?], and studied in Alexey Gronskiy's doctoral thesis [?].

Posterior agreement *validates algorithms* for solving stochastic optimization problems of the form

$$\min_c \mathbb{E}\left[R(c, X)\right].$$

Here, $R(c, X)$ is the result of evaluating a cost function on a candidate solution c on an instance of the problem described by the random variable X. The expectation is computed with X's distribution.

In posterior agreement, we assume that an algorithm is a function that computes, from a given observation X', a *posterior distribution* $p(\cdot \mid X')$ over a finite space $\mathcal{C}$ of feasible solutions. Every algorithm can be converted into such an algorithm, even if it is deterministic. In this case, the posterior distribution is just a distribution that assigns probability mass one to the algorithm's output.

Posterior agreement assesses the performance of an algorithm by computing the *expected log posterior agreement*:

$$\mathbb{E}_{X', X''} \log\left(|\mathcal{C}|\, \kappa\left(X', X''\right)\right), \tag{1}$$

where

$$\kappa\left(X', X''\right) := \sum_{c \in \mathcal{C}} p(\cdot \mid X') p(\cdot \mid X'').$$

The expected log posterior agreement requires the joint probability distribution of X' and X'', which is often unknown. One only has access to a handful of observations, at least two: X' and X''. In this case, one can use the *empirical log posterior agreement*:

$$\log\left(|\mathcal{C}|\, \kappa\left(X', X''\right)\right). \tag{2}$$

Moreover, the empirical log posterior agreement is intended to be a metric to compare different algorithms. Therefore, it is often sufficient to measure what is called the *posterior agreement kernel*

$$\kappa\left(X', X''\right).$$

We show how posterior agreement can compare, for example, Prim's, Kruskal's, and the reverse-delete algorithm for computing spanning trees for graphs whose edge weights

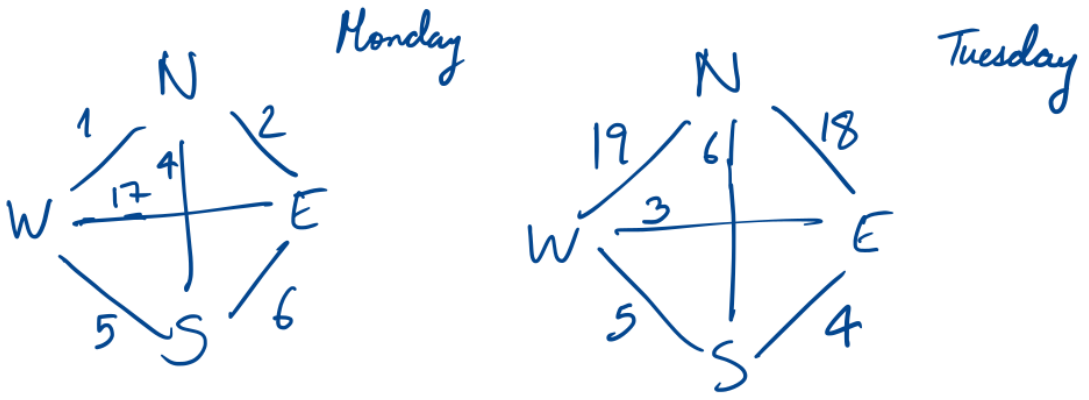


Figure 1: A graph with its observed edge weights for two different days.

are defined by random variables, as in the example above. Posterior agreement can also compare among different cost functions and hyperparameter values, when training machine learning models.

Therefore, posterior agreement advocates that, in the context of stochastic optimization, *algorithms should aim for maximizing the posterior agreement kernel, given two instances of the problem*, rather than minimizing the cost function for either instance or an aggregate of these instances.

2 Motivation

Figure 1 gives two graphs modeling a city with four main locations: North, East, South, and West. There is a road connecting any two different locations and, depending on the day, traversing that road by car takes some amount of time. The number labeling each edge indicates that time. Observe that the time varies with each day.

Consider the minimum spanning tree of each of these two graphs. For Monday's graph, the tree is the one with North at the root and all other locations as children of North. For Tuesday's graph, the tree is the one with South at the root and all other locations as children of South. We call these two trees the North and South trees, respectively.

Can we estimate from this information what Wednesday's graph's minimum spanning tree will look like?

The city graph can be understood as a random variable X. More precisely, it is a collection of 6 random variables, each of them defining an edge's weight on a particular day. Hence, the weight $R(c, X)$ of a tree c in X, which is the sum of the edge weights in c, is also a random variable. We can then formulate the problem of the minimum spanning tree for our case as the following stochastic optimization problem:

$$\arg\min_{c\in\mathcal{C}} \mathbb{E}\left[R(c, X)\right], \tag{3}$$

where $\mathcal{C}$ denotes all spanning trees of the city graph.

In practice, the main challenge in this type of optimization problems is that we do not know X's probability distribution. One natural way to go around this issue and approximately solve Problem 3 is to substitute $\mathbb{E}\left[R(c, X)\right]$ with an *empirical estimate*:

$$\mathbb{E}\left[R(c, X)\right] \approx \sum_{i\leq N} R(c, X_i),$$

where $\{X_1, \ldots, X_n\}$ is a sample of observed values of X. When n is sufficiently large and each X_i is independent from the others, then by the law of large numbers, this becomes a good approximation of $\mathbb{E}\left[R(c, X)\right]$. We call the *empirical risk minimizer* the solution of the problem:

$$\arg\min_{c \in \mathcal{C}} \sum_{i \leq n} \left[R(c, X_i)\right], \tag{4}$$

We present later scenarios where the empirical risk minimizer is not the best solution we can obtain, especially when we have only very few observations . One example is the problem of computing the minimum of an array $X = (X_1, \ldots, X_n)$ of random variables. Given just two observations X^1 and X^2, approximating $\min_i \mathbb{E}X_i$ with $\min_i X_i^1 + X_i^2$ is not the best we can do.

3 Overview

Posterior agreement originates from formalizing an algorithm $\mathcal{A}$ *as a communication channel* by which a sender and a receiver communicate outputs from $\mathcal{A}$. The robustness of an algorithm is measured by the *capacity* of that communication challenge, where the capacity defines the maximum number of distinguishable messages that can be communicated through the channel.

Shannon's channel coding theorem

Consider a channel by which a sender can transmit bits to a receiver. Assume that the channel is noisy in the sense that a bit can be flipped during transmission with probability $\epsilon < 0.5$. The sender and the receiver agree on transmitting only bitstrings of length n. We call these bitstrings *codewords.* Observe that if $\epsilon = 0$, then the sender can reliably communicate 2^n different messages to the receiver, by agreeing in advance with the receiver on a way to encode each message as one codeword of length n. This correspondence is called a *code.*

In practice, $\epsilon > 0$. Therefore, the sender and the receiver need to agree on a code that is robust to the channel's noise. We now show two examples of codes:

- They could agree on just 2 messages, encoded as $00\ldots0$ and $11\ldots1$, respectively. If the codeword length is sufficiently large, then with high probability, less than half of the bits will be flipped during transmission. As a result, the receiver can almost surely identify the message from the received codeword, by just counting the frequency of each bit in the codeword.
- They agree on sending 2^n messages, each encoded with a unique codeword of length n. With high probability, some of the bits will be flipped during transmission and the receiver will fail to identify the message that the sender tried to communicate.

Observe that these two codes represent two extremes, as $n \to \infty$. On one hand, the first code communicates only 2 messages, but with high probability of success. On the other hand, the second code communicates 2^n messages, but with low probability of success. One can also imagine other codes that strike a balance between the number of messages to be communicated and the success probability.

Shannon's coding theorem answers the following question. *What is the code that maximizes the number of messages that the sender can communicate to the receiver,* *while attaining a probability of success close to 1, as $n \to \infty$?* When $\epsilon > 0$, this number is clearly below 2^n, but since $\epsilon < 0.5$, this number must be positive. Shannon shows that this number is 2^{nc}, where c is *the channel's capacity*, a quantity that is defined by the channel. Therefore, channels with higher capacity allow the communication of more messages at the same codeword length.

Posterior agreement

Posterior agreement originates from modeling an algorithm $\mathcal{A}$ as a communication channel $C_{\mathcal{A}}$, where a sender communicates outputs from $\mathcal{A}$ to a receiver. We argue that the capacity of $C_{\mathcal{A}}$ is defined by $\mathcal{A}$'s robustness to noise in the input. That is, if $\mathcal{A}$ is robust to noise, then it is possible to communicate many more messages through $C_{\mathcal{A}}$ than when $\mathcal{A}$ is sensitive to noise.

We give an overview of this argument next. A rigorous argument is given in Section 5.

We assume given an *instance space* $\mathcal{X}$, comprising all possible observations, and a *solution space* $\mathcal{C}$, comprising all possible solutions. A phenomenon is then a probability distribution over $\mathcal{X}$. We assume that algorithms intending to solve Problem 3 receive in the input an observation X' and output a distribution $p(\cdot \mid X')$ over $\mathcal{C}$.

Example 1. In the minimum spanning tree problem, a codeword is a spanning tree, a phenomenon is a distribution governing a graph's edge weights, and an observation is a graph whose edge weights are drawn from a fixed distribution.

Example 2. In the centroid-based clustering problem, a codeword is a cluster assignment function and a set of centroids, an observation is a set of points to be clustered, and a phenomenon is a distribution where the points are drawn from.

For our analysis, we assume that each instance space contains all observations of a given "size" $n \in \mathbb{N}$. This size n is a notion that measures the observations' and phenomena's complexity. For example, in the minimum spanning tree problem, an instance space contains only all weighted graphs with a fixed number n of vertices.

For an algorithm $\mathcal{A}$, we define a communication channel $C_{\mathcal{A}}$ that works as follows. To use the channel, a sender picks an instance X', drawn from a phenomenon p_X, computes and inputs $p(\cdot \mid X')$ to the channel. The channel replaces $p(\cdot \mid X')$ with $p(\cdot \mid X'')$, where X'' is a fresh new instance drawn from p_X. The channel outputs $p(\cdot \mid X'')$ to the receiver.

We now emphasize the key insight of this modeling. If $\mathcal{A}$ is robust to the fluctuations in X', then there should not be much difference between $p(\cdot \mid X')$ and $p(\cdot \mid X'')$. In contrast, if $\mathcal{A}$ is very sensitive to the fluctuations in X', then $p(\cdot \mid X')$ and $p(\cdot \mid X'')$ may be very different. Hence, $\mathcal{A}$'s robustness to noise defines how many different "messages" can we send through this channel. We conclude then that $\mathcal{A}$'s robustness can be measured by the capacity of this channel $C_{\mathcal{A}}$. This capacity, as we show in Section 5, is measured by the expected log posterior agreement. For this reason, we argue that algorithms intended to solve Problem 3 shall be measured by their expected log posterior agreement.

Observe the following analogies to Shannon's coding theory. Channels have a capacity that define the maximum number of distinguishable messages that can be communicated. Channels with higher capacity are preferable, as they allow more different messages to be communicated. Analogously, we argue that algorithms can be modeled as channels and, therefore, have a capacity, which we later show to be the expected log posterior agreement. Hence, we argue that algorithms with higher expected log posterior agreement are preferable, as they allow more different messages to be communicated.

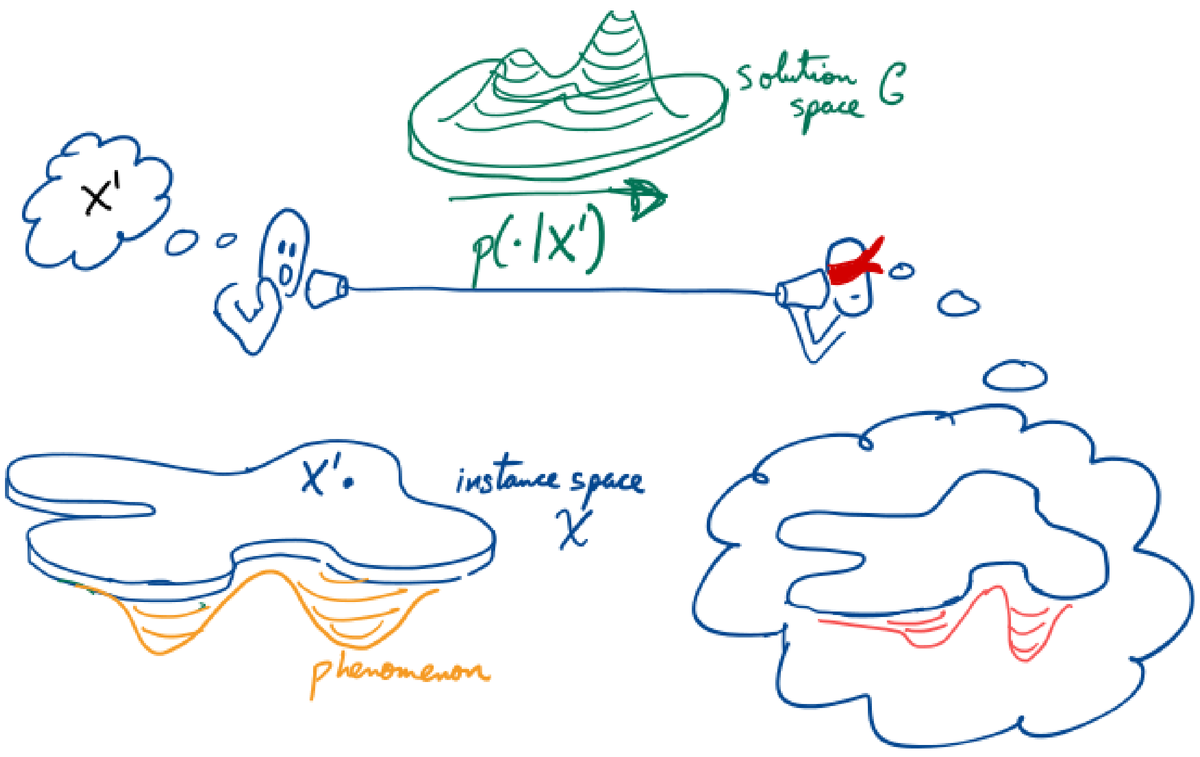


Figure 2

Protocol overview

The protocol describes a code by which a sender can communicate messages to a receiver. Let $\mathcal{A}$ be an algorithm intending to solve Problem 3. In our case, a message is a phenomenon and a codeword is the output $p(\cdot \mid X')$ of $\mathcal{A}$ when given an observation X' from a phenomenon as input. Analogous to Shannon's coding theorem, the sender and the receiver aim to maximize the number of different messages that the sender can communicate, while ensuring that the receiver's probability of success goes to 1 as $n \to \infty$.

Figure 2 gives an overview of the protocol. The sender must describe a phenomenon q to a receiver through a noisy channel. The sender makes an observation X' from q, uses $\mathcal{A}$ to compute $p(\cdot \mid X')$, and sends through the channel to the receiver. The channel is noisy and we represent its noise by replacing $p(\cdot \mid X')$ with $p(\cdot \mid X'')$, where X'' is another observation from q. The receiver succeeds if he is able, using $p(\cdot \mid X'')$, to distinguish X' from observations from other different phenomena.

We remark that this protocol is just a thought experiment. The protocol is computationally impossible for many interesting optimization problems. This is because the protocol requires that we know the underlying distribution behind the observations of one phenomenon. This is not possible in most of the cases. Nonetheless, the protocol provides a formal justification and motivation for posterior agreement.

Protocol example

We now give a more precise overview by showing how posterior agreement works with a simple example. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two algorithms that estimate the mean of a univariate distribution, given only a sample from that distribution. For example, $\mathcal{A}_1$ fits a Gaussian to the sample via maximum-likelihood estimation whereas $\mathcal{A}_2$ does the same, but only using only the sample maximum and minimum. Suppose that both algorithms output Gaussian distributions that indicate where they believe that the mean is.

The protocol works as follows. Fix $n \in \mathbb{N}$, which denote the size of all the observations in the instance space. In our case, n denotes the sample size. The sender and the receiver are given the algorithm under evaluation $\mathcal{A}_i$ and then they choose the size $m_n \in \mathbb{N}$ of the set of messages that they the sender will attempt to communicate. Recall that they want to choose m_n as large as possible. However, a large m_n increases the probability P_n that the receiver fails to recognize the message from the codeword transmitted by the sender. We later see that the best choice for m_n is defined by $\mathcal{A}_i$.

The protocol proceeds then as follows:

1. (Figure 3) The sender and the receiver agree on a set of $m := m_n$ phenomena, which we represent with the probability distributions $q_1, q_2, \ldots, q_m$.

2. (Figure 3) The sender and the receiver together make one observation X'_i of each phenomenon q_i, with $i \leq m$. In this case, an observation is a sample of points from q_i. Afterwards, they use $\mathcal{A}_i$ to compute a distribution $p(\cdot \mid X')$ for each observation X'.

3. (Figure 4) An observation X' is chosen out of these m observations uniformly at random. X' is given to the sender, but kept secret from the receiver.

4. (Figure 5) The sender sends $p(\cdot \mid X')$ to the receiver through a *noisy communication channel.* This channel replaces $p(\cdot \mid X')$ with $p(\cdot \mid X'')$, where X'' is a fresh new observation from the same phenomenon where X' comes from.

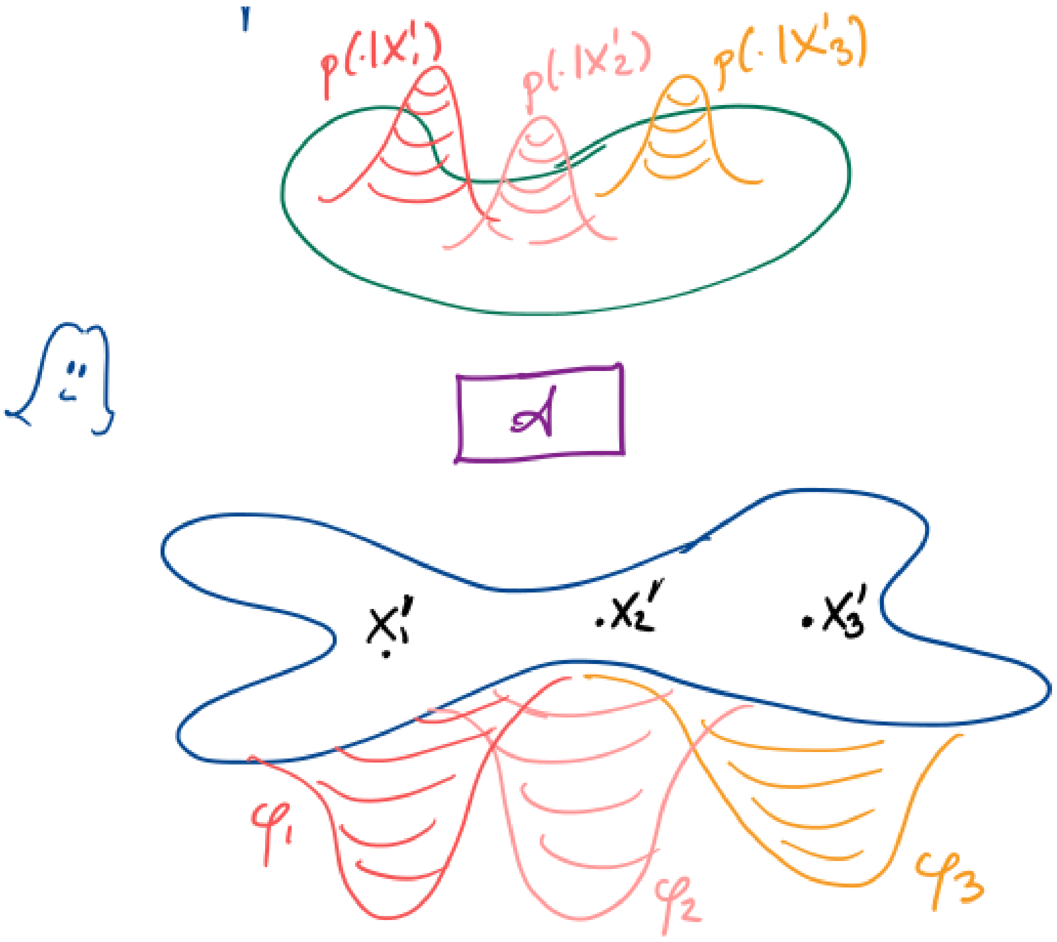
5. (Figure 6) The receiver gets $p(\cdot \mid X'')$ and must now guess which observation in $\{X'_1, \ldots, X'_m\}$ the sender chose in Step 3. For this, the receiver uses the natural approach of guessing the observation $\hat{X}$ for which $p(\cdot \mid \hat{X})$ overlaps the most with $p(\cdot \mid X'')$. In other words, the receiver guesses the observation $\hat{X}$ that fulfills:
$$\kappa\left(\hat{X}, X'\right) \geq \kappa\left(Y, X'\right), \text{ for all Y.}$$

6. If $\hat{X} = X'$, the receiver has succeeded.

Receiver's probability of failure

We show in Section 5.4 that the receiver's failure probability is bounded above by
$$\exp\left(-\mathbb{E}_{X',X''}\left[\log\left(|\mathcal{C}|\,\kappa\left(X', X''\right)\right)\right] + \epsilon \log|\mathcal{C}| + \log m\right),$$
where $\epsilon > 0$ is arbitrary, $\mathcal{C}$ is the solution space, and m is the number of observations defined in Step 1.

Assume now that $\log|\mathcal{C}| = \Omega(n)$. This is a reasonable assumption, as for many interesting problems, $\mathcal{C}$ grows exponentially on n. Observe that the algorithm can only influence $\kappa\left(X', X''\right)$. If ϵ is sufficiently small and the algorithm ensures that
$$\mathbb{E}_{X',X''} \log\left(|\mathcal{C}|\,\kappa\left(X', X''\right)\right) - \log m = \Omega(n),$$
then the receiver's failure probability becomes 0 as $n \to \infty$. The algorithm can ensure this by *maximizing the expected log posterior agreement.* The larger this quantity is, the higher m can be and the more messages the sender can communicate to the receiver.



$p(\cdot|X'_1)$
$p(\cdot|X'_2)$
$p(\cdot|X'_3)$
$\mathcal{A}$
X'_1
X'_2
X'_3
φ_1
φ_2
φ_3

Figure 3

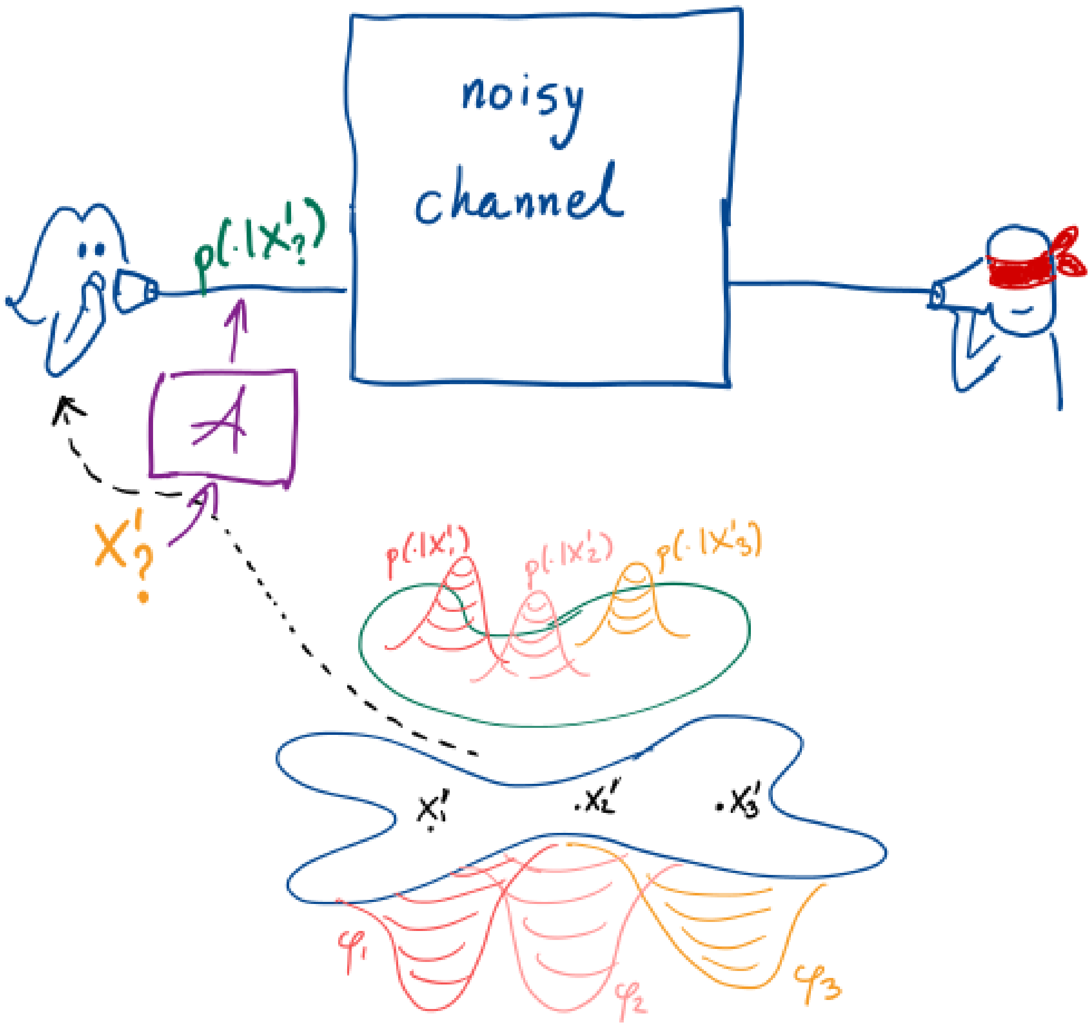

noisy
channel
$p(\cdot|X'_?)$
$\mathcal{A}$
$X'_?$
$p(\cdot|X'_1)$
$p(\cdot|X'_2)$
$p(\cdot|X'_3)$
X'_1
X'_2
X'_3
φ_1
φ_2
φ_3

Figure 4

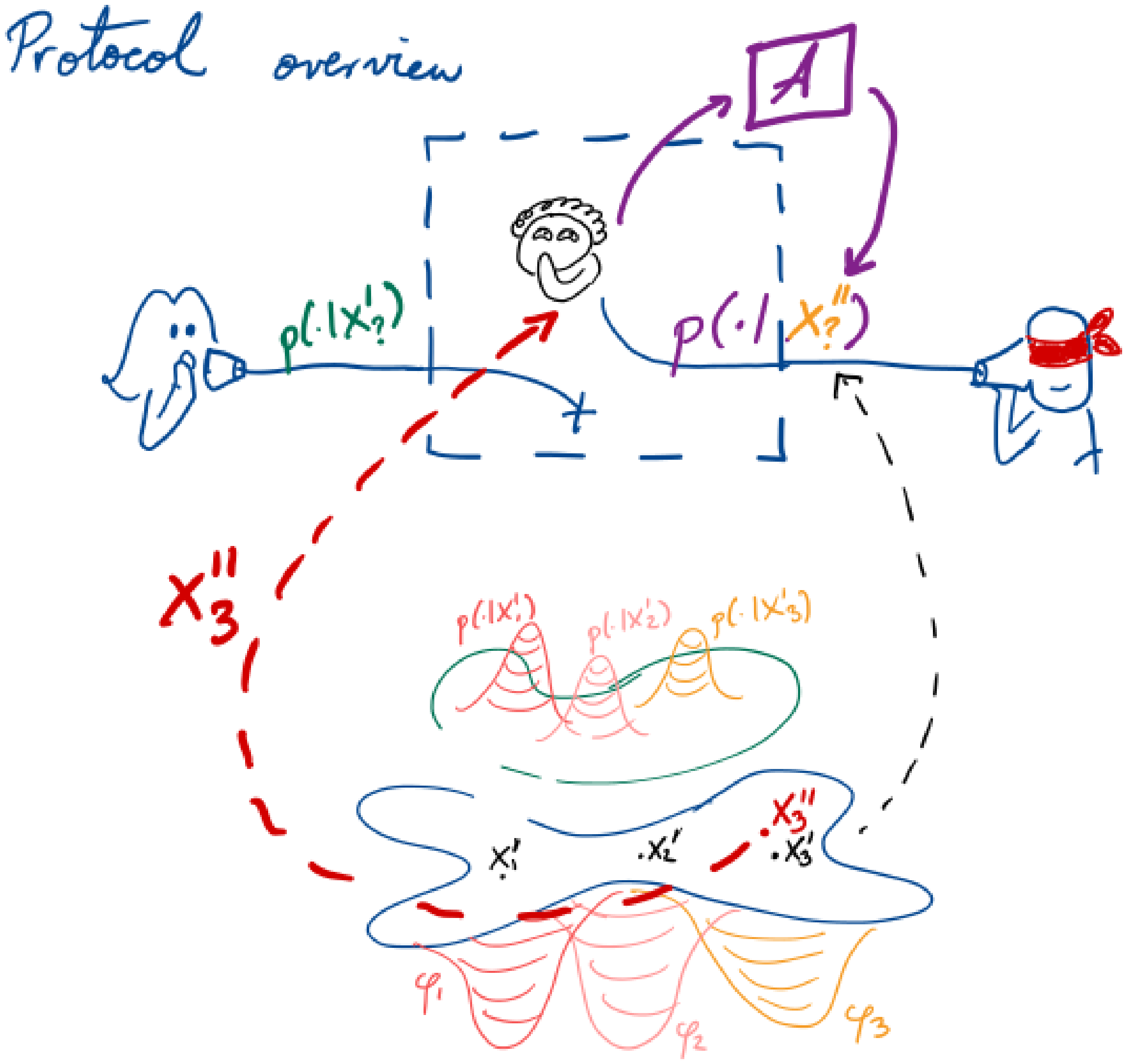

Protocol overview
A
$p(\cdot|X'_?)$
$p(\cdot|X''_?)$
X''_3
$p(\cdot|X'_1)$
$p(\cdot|X'_2)$
$p(\cdot|X'_3)$
X''_3
X'_1
X'_2
X'_3
φ_1
φ_2
φ_3

Figure 5

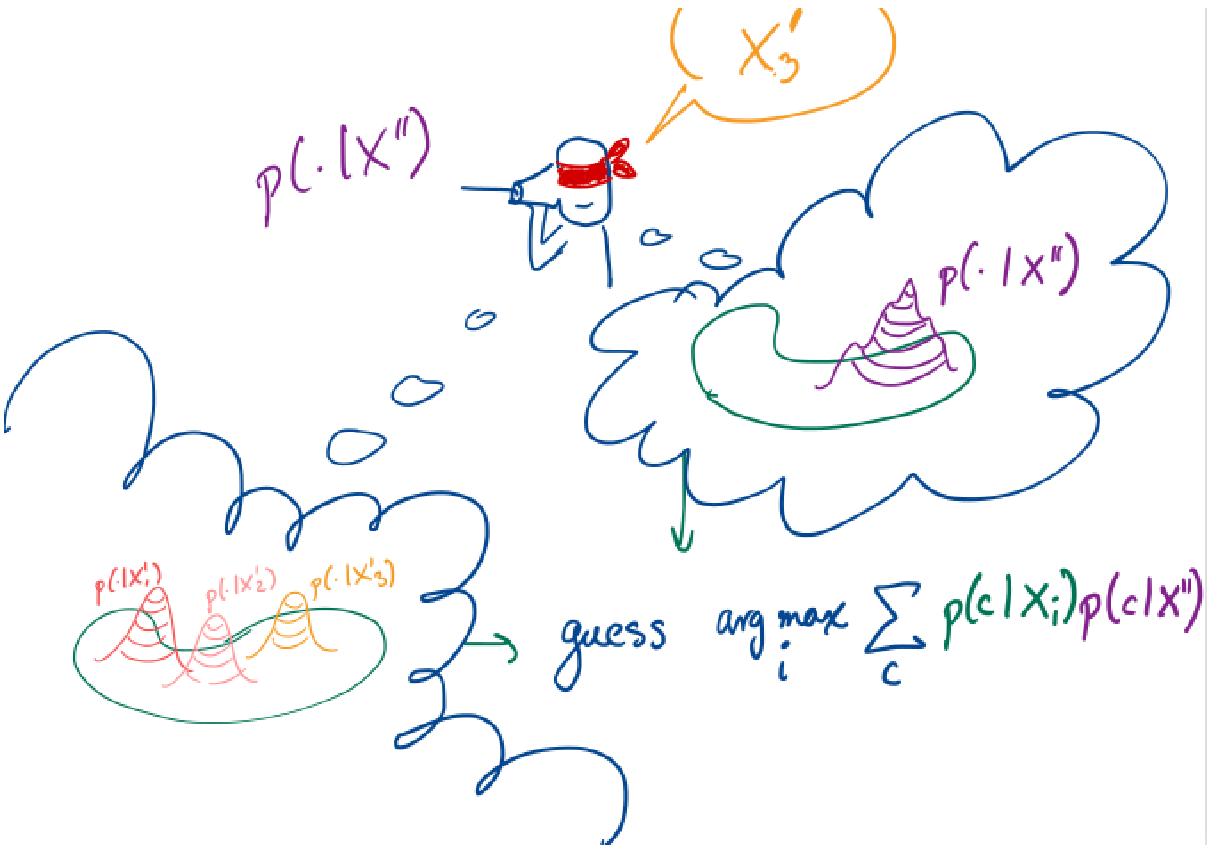


Figure 6

4 Shannon's channel coding theorem

To properly formulate and analyze the communication protocol above, we build upon Shannon's channel coding theorem. This theorem measures how much information we can optimally send through a communication channel. This section is mainly based on Chapter 7 from Thomas and Cover [?].

4.1 Channels

We understand a channel as a medium by which one sender can send symbols from a fixed set $\mathfrak{A}$ to a receiver. We allow channels to be noisy, meaning that the symbol a can be altered to another symbol b with probability $p(b \mid a)$ during the transmission through the channel. We do not allow the receiver to give feedback to the sender.

Definition 1. A *(noisy) channel* is a pair $\big(\mathfrak{A}, \{p(\cdot \mid a)\}_{a\in\mathfrak{A}}\big)$, where $\mathfrak{A}$ is a set and $p(\cdot \mid a)$, for $a \in \mathfrak{A}$, is a conditional distribution on $\mathfrak{A}$.

For convenience, we sometimes write just $\mathcal{P}$ to denote the family of distributions $\{p(\cdot \mid a)\}_{a\in\mathfrak{A}}$. Observe that we assume that transmissions are independent from each other. What the receiver gets does not influence what the user sends or the channel's conditional probabilities in the future.

Example 3. The *binary channel* is the channel with $\mathfrak{A} = \{0, 1\}$ and $p(b \mid a) = \mathbb{I}\{a = b\}$, for $a, b \in \mathfrak{A}$, where $\mathbb{I}$ is the indicator function. This is a channel where there is no noise interference.

Example 4. The *noisy binary channel* is the binary channel, but with

$$p(b \mid a) \begin{cases} 1-\epsilon & \text{if } a = b \text{ and} \\ \epsilon & \text{if } a \neq b. \end{cases}$$

Unless stated otherwise, we assume that $0 < \epsilon \leq 0.5$.

Example 5. The *typewriter channel* is the channel with $\mathfrak{A} = \{\mathtt{a}, \mathtt{b}, \dots, \mathtt{z}\}$ and $p(b \mid a) = \mathbb{I}\{a = b\}$, for $a, b \in \mathfrak{A}$, where $\mathbb{I}$ is the indicator function.

Example 6. The *noisy typewriter channel* is the channel with $\mathfrak{A} = \{\mathtt{a}, \mathtt{b}, \dots, \mathtt{z}\}$, but with

$$p(b \mid a) \begin{cases} 1-\epsilon & \text{if } a = b, \\ \epsilon & \text{if } a = b + 1 \bmod 26. \end{cases}$$

4.2 Channel capacity

Which of the channels above sends *the most information per transmission*? Intuitively, a letter has more information than a bit and the presence of noise affects the amount of information we send in one transmission. Indeed, sending one letter through the typewriter channel provides more information than sending one letter through a noisy typewriter channel. Also, a letter has more information than a bit. Hence, we say that the typewriter channel has *the most capacity*: one transmission through this channel carries in average more information than one transmission through any of the other three channels. Similarly, we say that the noisy binary channel has *the least capacity.*

We now formally define channel capacity.

Definition 2. For a channel $\big(\mathfrak{A}, \{p(\cdot \mid a)\}_{a \in \mathfrak{A}}\big)$, its capacity is

$$\max_{p(\cdot)} I(S; \hat{S}),$$

where S and $\hat{S}$ are random variables denoting a symbol input to the channel and the output symbol, respectively, when S is distributed according to $p(\cdot)$. We refer to $I(S; \hat{S})$ as the channel's *input-output mutual information* and the joint distribution of S and $\hat{S}$ as the *input-output distribution.*

Example 7. For the binary channel, we can reliably send one bit per transmission. This corresponds to the channel's capacity,

$$\max_{p(\cdot)} I(S; \hat{S}) = \max_{p(\cdot)} \{H(S) - H(\hat{S} \mid S)\} = \max_{p(\cdot)} H(S) = \max_{p(\cdot)} H(S) = 1.$$

The second equality follows from the fact that $\hat{S} = S$, so $H(\hat{S} \mid S) = 0$. The last equality follows from the fact that the distribution that maximizes the entropy of a Bernoulli random variable is the uniform distribution, which yields an entropy of 1 bit.

Example 8. A similar line of reasoning shows that the typewriter channel's capacity is $\max_{p(\cdot)} H(S) = \log 26 \approx 4.7$. This corresponds to the intuition that we can reliably send one letter per transmission, which contains around 4.7 bits of information.

Example 9. Consider now the noisy typewriter with $\epsilon = 0.5$. How many bits can we reliably send per transmission? Observe that one way to reliably send information is by agreeing to only send letters at even positions in the alphabetic order. In that way, if you receive, for example, `a` or `b`, you know for sure that the sender input `a` to the channel. However, by sending only the "even" letters, you need to double the efforts with respect to the typewriter channel without noise. As a consequence, the noisy typewriter has less capacity. One can actually show that $\max_{p(\cdot)} I(S; \hat{S}) = -1 + \log 26$, where a maximizing $p(\cdot)$ is the uniform distribution over the "even" letters.

4.3 Codes

Intuition on codes and rates

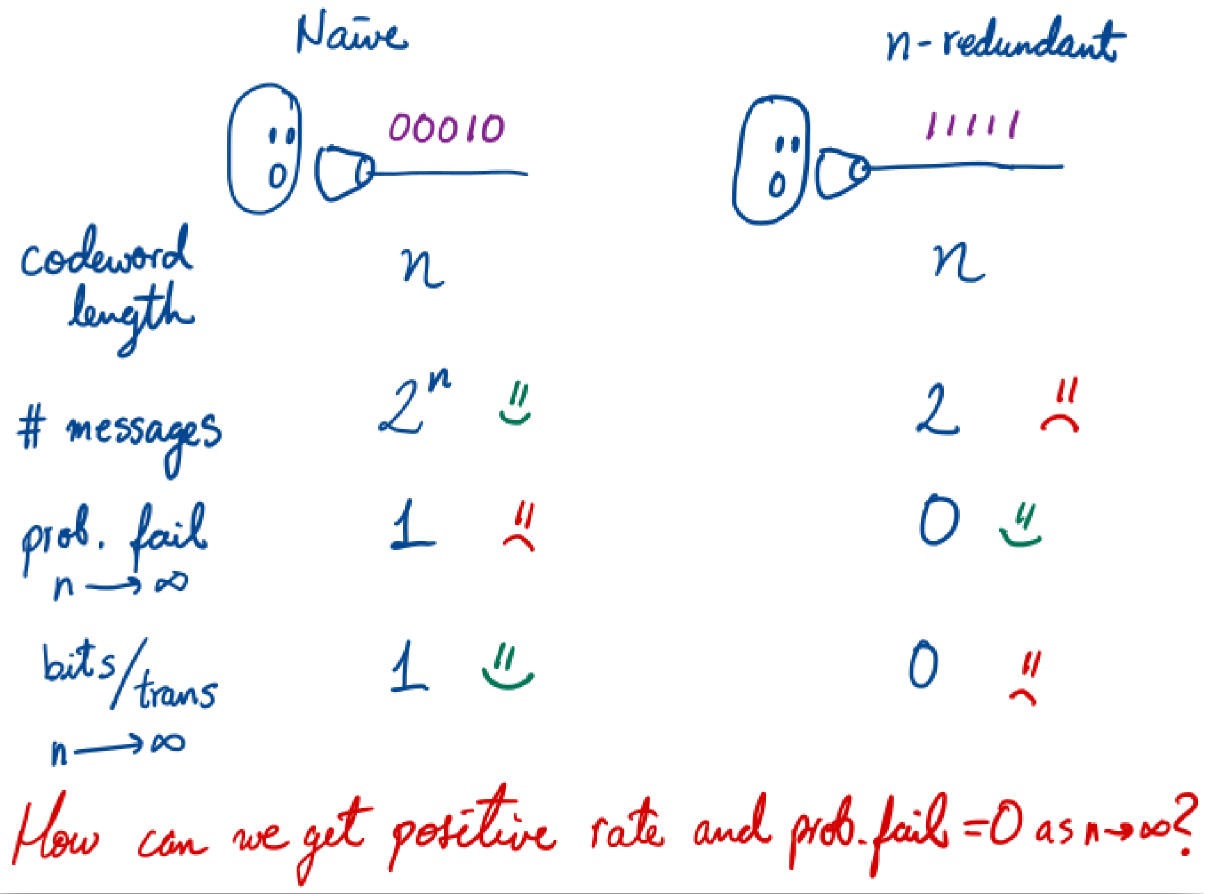
Consider the noisy binary channel. By sending several bits in a specific pattern to the channel, one can come up with sophisticated ways to transmit complex information through the channel, like images or spreadsheets. We illustrate this by showing how to use the binary channel to send letters in $\{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$. The sender and the receiver must first agree on a *code* for those letters. One such code, which we call *naïve code*, encodes the letter `a` as the *codeword* `00000`, `b` as `00001`, and so on. That is, the codeword for the i-th alphabet letter is number i in base 2, written as a bit string of length 5. In a similar fashion, we can conceive codes for communicating more complex data like images and spreadsheets.

Unfortunately, the code mentioned in the previous paragraph is sensitive to the channel's noise. If we send the codeword for `a`, the receiver may get the codeword for `b` with probability $\epsilon\,(1-\epsilon)^4$, yielding a *communication error.* Information theory has came up with smarter codes that reduce the probability of such a communication error, but at the cost of longer codewords. For example, we can use a code, which we call *the 5-redundant code.* This code encodes `a` as `00000` and `b` as `11111`. The receiver would then take the received codeword $\hat{w}$ and search for the codeword w that closest codeword with respect to the Hamming distance. The receiver would then assume that the sender sent the letter associated to w. For example, if the receiver gets `11010`, then she assumes that the sender sent `11111`, which is the codeword for `b`. This code is more robust to noise than the naïve alphabet code. In comparison with the naïve code, more bits need to be flipped by noise in order to get a communication error. This is less likely than having one bit flipped in the naïve code's codewords.

Unfortunately, the robustness comes at the price of less messages. If we use codewords of length n only. Using the naïve code, the sender can communicate 2^n different messages. However, using the n-redundant code, the sender can communicate only 2. Assuming that the noise does not cause a communication error, we manage to transmit at a *rate* of one bit per transmission in the naïve code and one bit per n transmissions in the n-redundant code. In the limit, as $n \to \infty$, the naïve code attains a rate of 1 bit per transmission, but a probability of a communication error equal to 1. On the other hand, the n-redundant code attains a rate of 0 bits per transmission, but a probability of a communication error equal to 0. This comparison is summarized in Figure 7

Intuition on Shannon's channel coding theorem

We started by using the binary noisy channel to send bits and we are now devising codes to send a finite set of letters through the binary noisy channel. In a similar manner, we


00010
11111

	Naïve		n-redundant	
codeword length	n		n	
# messages	2^n	🙂	2	🙁
prob. fail $n \to \infty$	1	🙁	0	🙂
bits/trans $n \to \infty$	1	🙂	0	🙁

How can we get positive rate and prob. fail $= 0$ as $n \to \infty$?

Figure 7

can devise codes to send words through the binary noisy channel, by creating a code that maps words to bit strings of fixed length.

We can continue in this way to create codes to send images of a fixed size, videos of a fixed length, and so on. All this is done by using longer codewords.

Shannon's coding theorem concerns with the problem of finding sustainable strategies for building codes. By sustainable, we mean that the strategy should yield codes that fulfill two conditions.

- First, the codes attain and maintain a positive rate r of bits of information per transmission as $n \to \infty$.
- Second, the probability of a communication error goes to zero as $n \to \infty$.

Shannon's channel coding theorem states that there is a sustainable strategy for building codes, as long as the targeted rate r is below the channel's capacity. For a fixed codeword length n, the maximum number of messages that can be communicated with this strategy is $\lfloor 2^{nr} \rfloor$.

Formalization

For the definitions below, let $M, n \in \mathbb{N}$ and let $(\mathfrak{A}, \mathcal{P})$ be a channel.

Definition 3. An *(M, n)-code* is a pair (Enc, Dec) of functions with $Enc : \{1, 2, \ldots, M\} \to \mathfrak{A}^n$ and $Dec : \mathfrak{A}^n \to \{1, 2, \ldots, M\}$.

Definition 4. The *rate of a (M, n)-code* is

$$\frac{\log M}{n}.$$

Observe that if a (M, n)-code has rate r, then $M = 2^{nr}$.

Definition 5. For an (M, n)-code (Enc, Dec), its *probability of a communication error* is

$$\frac{1}{M} \sum_{i \leq M} \mathbb{P}\left(Dec(\hat{W}) \neq i \mid W = Enc\,(i) \right),$$

where $\mathbb{P}\left(Dec(\hat{W}) \neq i \mid W = Enc\,(i) \right)$ is the probability that the receiver decodes something different to i, given that we sent $Enc\,(i)$ through the channel.

Intuitively, the probability of a communication error is the probability that the receives decodes a wrong message when we send him the codeword of a message chosen uniformly at random.

Example 10. The n-redundant code discussed above is an example of a $(2^n, n)$-code, whose rate is $\log 2^n / n = 1$.

Definition 6. A rate r is *attainable* if there is a sequence of $(\lfloor 2^{nr} \rfloor, n)$-codes, indexed by n, such that the probability of a communication error goes to zero as $n \to \infty$.

4.4 Shannon's coding theorem

Typicality

We now recall some important notions from Thomas and Cover [**?**]

Theorem 1. (The asymptotic equipartition property) Let $S, S_1, S_2, \ldots, S_n$ be identically and independently distributed random variables with distribution $p(\cdot)$ over a space $\mathcal{S}$, then

$$-\frac{1}{n}p(S_1, \ldots, S_n) \to H(S), \quad \text{in probability as } n \to \infty.$$

This theorem follows from the weak law of large numbers. In our context, $S, S_1, S_2, \ldots, S_n$ denote symbols from a channel's alphabet and $p(\cdot)$ is $\arg\max_p I(S; \hat{S})$. That is, $p(\cdot)$ is the distribution that achieves the channel's capacity.

Definition 7. For $n \in \mathbb{N}$ and $\epsilon > 0$, the *typical set* $A_\epsilon^{(n)}$ with respect to $p(\cdot)$ is the set of sequences $(s_1, \ldots, s_n) \in \mathcal{S}^n$ such that

$$H(S) - \epsilon \leq -\frac{1}{n}\log p(s_1, s_2, \ldots, s_n) \leq H(S) + \epsilon.$$

A sequence in $A_\epsilon^{(n)}$ is called a *typical sequence.*

In our context, codewords will consist of typical sequences. The next theorem justifies the following intuitions:

1. All typical sequences have approximately the same probability $\approx 2^{-nH(S)}$.
2. If you draw a sequence in $\mathcal{S}^n$, using $p(\cdot)$, then the resulting sequence is typical with high probability.
3. $\left|A_\epsilon^{(n)}\right| \approx 2^{nH(S)}$.

Theorem 2.

1. If $(s_1, s_2, \ldots, s_n) \in A_\epsilon^{(n)}$, then $2^{-n(H(S)+\epsilon)} \leq p(s_1, \ldots, s_n) \leq 2^{-n(H(S)-\epsilon)}$.
2. $\mathbb{P}\left(A_\epsilon^{(n)}\right) > 1 - \epsilon$, for sufficiently large n.
3. $(1-\epsilon)2^{n(H(S)-\epsilon)} \leq \left|A_\epsilon^{(n)}\right| \leq 2^{n(H(S)+\epsilon)}$.

The reader can take it as an exercise to proof these claims. The proofs are in Thomas and Cover [**?**].

Definition 8. Let $n \in \mathbb{N}$ and let $p_{S\hat{S}}(\cdot, \cdot)$ be the joint distribution of two random variables S and $\hat{S}$, whose ranges are $\mathcal{S}$ and $\hat{\mathcal{S}}$, respectively. The set $A_\epsilon^{(n)}$ of *jointly typical sequences* with respect to $p_{S\hat{S}}$ is the set of pairs $(\mathbf{s}^n, \hat{\mathbf{s}}^n)$ of sequences that fulfill the following:

1. $\left|-\frac{1}{n}\log p_{S^n}(\mathbf{s}^n) - H(S)\right| < \epsilon$.
2. $\left|-\frac{1}{n}\log p_{\hat{S}^n}(\hat{\mathbf{s}}^n) - H(\hat{S})\right| < \epsilon$.

3. $\left| -\frac{1}{n} \log p_{S^n \hat{S}^n}(\mathbf{s}^n, \hat{\mathbf{s}}^n) - H(S, \hat{S}) \right| < \epsilon.$

A pair in $A_\epsilon^{(n)}$ is called a *jointly typical pair of sequences.*

We clarify that, for $\mathbf{s}^n = (s_1, \ldots, s_n)$ and $\hat{\mathbf{s}}^n = (\hat{s}_1, \ldots, \hat{s}_n)$,

$$p_{S^n}(\mathbf{s}^n) = \prod_{i \le n} p_S(s_i),$$

$$p_{\hat{S}^n}(\hat{\mathbf{s}}^n) = \prod_{i \le n} p_{\hat{S}}(\hat{s}_i) = \prod_{i \le n} \sum_s p_S(s) p_{\hat{S}|S}(\hat{s}_i \mid s), \text{and}$$

$$p_{S^n \hat{S}^n}(\mathbf{s}^n, \hat{\mathbf{s}}^n) = \prod_{i \le n} p_{S\hat{S}}(s_i, \hat{s}_i) = \prod_{i \le n} p_S(s_i) p_{\hat{S}|S}(\hat{s}_i \mid s_i).$$

In the context of communication via a channel, $p_{S^n \hat{S}^n}$ represents the joint distribution of $\mathbf{S}^n$ and $\hat{\mathbf{S}}^n$, where

- $\mathbf{S}^n$ denotes a random codeword, where each symbol was chosen at random according to the distribution $p_S = \arg\max_p I(S; \hat{S})$ that achieves channel capacity.
- $\hat{\mathbf{S}}^n$ denotes a codeword that the channel would output, after we send $\mathbf{S}^n$ as input.

We call $p_{S^n \hat{S}^n}$ the *codeword input-output distribution.*

In the context of communication via a channel, the following theorem justifies the following intuitions:

1. Suppose that we build a codeword $\mathbf{s}^n$ at random by choosing each of its symbols at random according to p_S, the distribution that attains channel capacity. Then we send $\mathbf{s}^n$ through the channel and let $\hat{\mathbf{s}}^n$ be the output codeword. Then $(\mathbf{s}^n, \hat{\mathbf{s}}^n)$ is jointly typical with high probability.

2. Suppose now that we build another codeword $\mathbf{q}^n$ at random using the same procedure. Then it is very *unlikely* that $(\mathbf{q}^n, \mathbf{y}^n)$ is jointly typical.

Theorem 3.

1. $\mathbb{P}\left(\left(\mathbf{S}^n, \hat{\mathbf{S}}^n \right) \in A_\epsilon^{(n)} \right) \to 1$, as $n \to \infty$.

2. If $\mathbf{Q}^n \sim p_{S^n}(\cdot)$ and $\hat{\mathbf{S}}^n \sim p_{\hat{S}^n}(\cdot)$ (i.e., they are drawn independently at random from the marginal distributions of $p_{S^n \hat{S}^n}$), then

$$(1-\epsilon) 2^{-n\left(I(S;\hat{S}) + 3\epsilon\right)} \le \mathbb{P}\left(\left(\mathbf{Q}^n, \hat{\mathbf{S}}^n \right) \in A_\epsilon^{(n)} \right) \le 2^{-n\left(I(S;\hat{S}) - 3\epsilon\right)}$$

These two intuitions justify the effectiveness of a very simple code, called *Shannon's random code.*

Shannon's random code

Theorem 4. A rate is attainable iff it is below the channel's capacity.

We only focus here on proving the following direction: if a rate is below the channel's capacity, then it is attainable, as it illustrates how to propose a $(\lfloor 2^{nr} \rfloor, n)$-code for communicating $\lfloor 2^{nr} \rfloor$ messages.

To prove this, we present, for $n > 1$, a $(\lfloor 2^{nr} \rfloor, n)$-code whose probability of a communication error is at most $p_n = 2^{-n(cap-3\epsilon-r)}$, where ϵ is chosen to be sufficiently small. Hence, if $r < cap$, we get that the probability of a communication error goes to zero as $n \to \infty$.

The code's encoder function *Enc* is defined as follows. For a message $m \leq \lfloor 2^{nr} \rfloor$, we define $Enc(m)$ as a string in $\mathbf{s}^n$ where each symbol was drawn from a distribution $p^* = \arg\max_{p(\cdot)} I(S; \hat{S})$. That is, a distribution that maximizes the channel's input-output mutual information and attains the channel's capacity.

The code's decoder function *Dec* is defined as follows. Given the string $\mathbf{s}^n$ output by the channel, *Dec* goes through each message m and tests if $(Enc(m), \mathbf{s}^n)$ is jointly typical with respect to the codeword input-output distribution $p_{S^n \hat{S}^n}$. *Dec* outputs the first message for which this test succeeds. If no message succeeds on the test, then *Dec* outputs an arbitrary message.

Theorem 5. The probability $\mathbb{P}(\mathcal{E})$ of a communication error for Shannon's random code goes to 0 as $n \to \infty$.

Proof. Let $\mathcal{K}$ be a random variable representing a possible code. Then

$$\mathbb{P}(\mathcal{E}) = \sum_{\mathcal{K}} \mathbb{P}(\mathcal{K})\, P_e(\mathcal{K}),$$

where $P_e(\mathcal{K})$ is the probability of a communication error for code $\mathcal{K}$. Observe now that

$$\begin{aligned}
\mathbb{P}(\mathcal{E}) &= \sum_{\mathcal{K}} \mathbb{P}(\mathcal{K})\, P_e(\mathcal{K}) \\
&= \sum_{\mathcal{K}} \mathbb{P}(\mathcal{K}) \frac{1}{\lfloor 2^{nr} \rfloor} \sum_{w \leq \lfloor 2^{nr} \rfloor} \mathbb{P}\left(Dec(\hat{\mathbf{S}}^n) \neq w \mid \mathbf{S}^n = Enc(w)\right) \\
&= \frac{1}{\lfloor 2^{nr} \rfloor} \sum_{\mathcal{K}} \sum_{w \leq \lfloor 2^{nr} \rfloor} \mathbb{P}(\mathcal{K})\, \mathbb{P}\left(Dec(\hat{\mathbf{S}}^n) \neq w \mid \mathbf{S}^n = Enc(w)\right).
\end{aligned}$$

Observe now that all codewords were chosen independently at random, so

$$\mathbb{P}\left(Dec(\hat{\mathbf{S}}^n) \neq w \mid \mathbf{S}^n = Enc(w)\right) = \mathbb{P}\left(Dec(\hat{\mathbf{S}}^n) \neq 1 \mid \mathbf{S}^n = Enc(1)\right),$$

for $w > 1$. Hence,

$$\begin{aligned}
\mathbb{P}(\mathcal{E}) &= \frac{1}{\lfloor 2^{nr} \rfloor} \sum_{\mathcal{K}} \sum_{w \leq \lfloor 2^{nr} \rfloor} \mathbb{P}(\mathcal{K})\, \mathbb{P}\left(Dec(\hat{\mathbf{S}}^n) \neq w \mid \mathbf{S}^n = Enc(w)\right) \\
&= \sum_{\mathcal{K}} \mathbb{P}(\mathcal{K})\, \mathbb{P}\left(Dec(\hat{\mathbf{S}}^n) \neq 1 \mid \mathbf{S}^n = Enc(1)\right) \\
&= \mathbb{P}(\mathcal{E} \mid w = 1).
\end{aligned}$$

This means that the probability of a communication error is equal to the probability of a communication error, assuming that the sender sent the codeword for message 1.

Note that, in Shannon's random code, the event of a communication error implies at least one of the following events: the received codeword $\hat{\mathbf{S}}^n(1)$ is not jointly typical with the codeword $\mathbf{S}^n(1)$ for message 1 or the received codeword $\hat{\mathbf{S}}^n(1)$ is jointly typical with the codeword $\mathbf{S}^n(w)$ for a message $w > 1$. More precisely,

$$\mathbb{P}(\mathcal{E} \mid M = 1) = \mathbb{P}\left(\begin{array}{c} \left(\mathbf{S}^n(1), \hat{\mathbf{S}}^n(1)\right) \notin A_\epsilon^{(n)} \text{ or} \\ \left(\mathbf{S}^n(2), \hat{\mathbf{S}}^n(1)\right) \in A_\epsilon^{(n)} \text{ or} \\ \left(\mathbf{S}^n(3), \hat{\mathbf{S}}^n(1)\right) \in A_\epsilon^{(n)} \text{ or} \\ \vdots \\ \left(\mathbf{S}^n(\lfloor 2^{nr} \rfloor), \hat{\mathbf{S}}^n(1)\right) \in A_\epsilon^{(n)}. \end{array}\right).$$

By the union bound,

$$\mathbb{P}(\mathcal{E} \mid M = 1) \le \mathbb{P}\left(\left(\mathbf{S}^n(1), \hat{\mathbf{S}}^n(1)\right) \notin A_\epsilon^{(n)}\right) + \sum_{w>1} \mathbb{P}\left(\left(\mathbf{S}^n(w), \hat{\mathbf{S}}^n(1)\right) \in A_\epsilon^{(n)}\right).$$

We now apply Theorem 3, which implies the following:

- $\mathbb{P}\left(\left(\mathbf{S}^n(1), \hat{\mathbf{S}}^n(1)\right) \notin A_\epsilon^{(n)}\right) \to 1$, as $n \to \infty$. In other words, $\mathbb{P}\left(\left(\mathbf{S}^n(1), \hat{\mathbf{S}}^n(1)\right) \notin A_\epsilon^{(n)}\right) \to 0$, as $n \to \infty$.
- $\mathbb{P}\left(\left(\mathbf{S}^n(w), \hat{\mathbf{S}}^n(1)\right) \in A_\epsilon^{(n)}\right) \le 2^{-n\left(I(S;\hat{S})-3\epsilon\right)}$. This is because, for $w > 1$, $\mathbf{S}^n(1)$ and $\mathbf{S}^n(w)$ were independently drawn from p_{S^n} and, therefore, $\hat{\mathbf{S}}^n(1)$ and $\mathbf{S}^n(w)$ were independently drawn from $p_{\hat{S}^n}$ and p_{S^n}, respectively.

Using these observations we get that

$$\begin{aligned}
\mathbb{P}(\mathcal{E} \mid M = 1) &\le \mathbb{P}\left(\left(\mathbf{S}^n(1), \hat{\mathbf{S}}^n(1)\right) \notin A_\epsilon^{(n)}\right) + \sum_{w>1} \mathbb{P}\left(\left(\mathbf{S}^n(w), \hat{\mathbf{S}}^n(1)\right) \in A_\epsilon^{(n)}\right) \\
&\le \mathbb{P}\left(\left(\mathbf{S}^n(1), \hat{\mathbf{S}}^n(1)\right) \notin A_\epsilon^{(n)}\right) + \sum_{w>1} 2^{-n\left(I(S;\hat{S})-3\epsilon\right)} \\
&= \mathbb{P}\left(\left(\mathbf{S}^n(1), \hat{\mathbf{S}}^n(1)\right) \notin A_\epsilon^{(n)}\right) + \left(\lfloor 2^{nr} \rfloor - 1\right) 2^{-n\left(I(S;\hat{S})-3\epsilon\right)} \\
&\le \mathbb{P}\left(\left(\mathbf{S}^n(1), \hat{\mathbf{S}}^n(1)\right) \notin A_\epsilon^{(n)}\right) + 2^{nr} 2^{-n\left(I(S;\hat{S})-3\epsilon\right)} \\
&= \mathbb{P}\left(\left(\mathbf{S}^n(1), \hat{\mathbf{S}}^n(1)\right) \notin A_\epsilon^{(n)}\right) + 2^{-n\left(I(S;\hat{S})-r-3\epsilon\right)}.
\end{aligned}$$

Observe that we chose p_S as $\arg\max_p I(S;\hat{S})$, so $I(S;\hat{S}) = cap$, the channel's capacity. So, if the rate r is below the channel's capacity and ϵ is sufficiently small, then $\mathbb{P}(\mathcal{E} \mid M = 1) \to 0$ as $n \to \infty$. □

5 Communication protocol for algorithm validation

We now formalize the communication protocol where posterior agreement originates. We first explain an *ideal variant* where we know X's probability distribution p_X. Furthermore, we assume that p_X is from a parameterized family $\mathfrak{P}$ of probability distributions.

Afterwards, we explain the *empirical variant* where we do not know anything about p_X and, even worse, we only have *two observations* X' and X'' drawn from p_X. Recall that, in practice, we only have access to observations and have no information of the nature of the distribution where these observations came from.

5.1 Assumptions

For our statements to hold, we make the following assumptions.

Exponential solution space: We assume that $\mathcal{C}$ is discrete and that $\log|\mathcal{C}| = \Theta(n)$, where n measures the "size" of an observation from a phenomenon. For example, n measures the number of edges in a graph or the number of datapoints in a clustering instance. Intuitively, we assume that the solution space's size grows exponentially in n.

Probabilistic outputs: We assume that any algorithm for stochastic optimization, when given an input X' from an instance space $\mathcal{X}$, outputs a distribution $p(\cdot \mid X')$ over a discrete solution space $\mathcal{C}$. Every algorithm can be thought to be as such, even when it only outputs a fixed value $c_{X'} \in \mathcal{C}$, when given X' as input. In this case, $p(c \mid X') = 1$ if $c = c_{X'}$ and 0 otherwise.

5.2 Ideal variant

Let $\mathcal{A}$ be an algorithm intended for solving Problem 3. Posterior agreement originates from a communication protocol between a sender and a receiver. The sender must communicate, using $\mathcal{A}$'s outputs, the nature of a phenomenon to the receiver. The receiver must be able, using the information sent from the sender, to identify the phenomenon. The communication from the sender to the receiver is done through a noisy channel, whose noise is defined by $\mathcal{A}$ and the phenomenon.

Messages Fix $n \in \mathbb{N}$. Present the instance space $\mathcal{X}$ to the sender and the receiver. Then agree draw at random a set $\mathcal{F} \subseteq \mathfrak{P}$ of m phenomena and, for each $p \in \mathcal{F}$, draw an observation X' of size n. Let $\mathcal{M} = \{X'_1, \ldots, X'_m\}$. Agreeing on $\mathcal{F}$ is often not possible in practice, so we explain in the empirical variant how to deal with this. This $\mathcal{M}$ constitutes the set of possible messages that the sender may try to communicate to the receiver.

For the moment, we leave the value of m undefined. We leave for later to figure out what is the maximum value of m that we can use. The choice of m must still ensure that the probability of a communication error goes to 0, as $n \to \infty$.

Code Present algorithm $\mathcal{A}$ to the sender and the receiver. For each $X' \in \mathcal{M}$, use $\mathcal{A}$ to compute $p(\cdot \mid X')$. Define the code $(Enc_{\mathcal{A}}, Dec_{\mathcal{A}})$ as follows:

- $Enc_{\mathcal{A}}$ encodes $X' \in \mathcal{M}$ as the codeword $p(\cdot \mid X')$.
- $Dec_{\mathcal{A}}$ decodes a probability distribution $p(\cdot \mid Y)$ over $\mathcal{C}$ as any $\hat{X} \in \mathcal{M}$ such that $k\left(Y, \hat{X}\right) \geq k(Y, X)$, for all $X \in \mathcal{M}$. Here,

$$k(Y, X') := \sum_{c} p(c \mid Y) p(c \mid X').$$

Channel The channel is the pair $\left(\{p(\cdot \mid X')\}_{X' \in \mathcal{M}}, \mathcal{P}\right)$ where $\mathcal{P}$ is defined by the following probabilistic procedure. Assume that the sender inputs $p(\cdot \mid X')$ to the channel. The channel then replaces X' with a fresh new observation X'' from the same phenomenon where X' comes from. Then it uses the algorithm $\mathcal{A}$ to compute $p(\cdot \mid X'')$ and outputs that to the receiver.

Communication A message X' is selected uniformly at random from $\mathcal{M}$ and without the receiver's knowledge. The sender then sends the codeword $p(\cdot \mid X')$ through the channel. The receiver gets $p(\cdot \mid X'')$ and uses the decoding function to guess which message the sender sent. The receiver succeeds by correctly guessing X'.

Observe how the algorithm and the phenomenon define the channel's noise. Hence, the algorithm can be evaluated by *the capacity* of the resulting channel. This capacity is measured by its maximum attainable rate, which is obtained by figuring out how to maximize the number m of messages that can be used in the protocol, while making the probability of a communication error go to zero as n, the size of the observations, go to infinity. We carry this analysis in the empirical variant.

5.3 Empirical variant

The capacity of the channel built above cannot be computed, as we often only have access to a set of observations and not to the phenomena behind them. This means that we cannot compute the set $\mathcal{M}$ of messages, as described in the ideal variant. For this reason, we use an empirical variant, where we assume that we are given at least two observations X' and X'' of a same phenomenon.

Messages (Figure 8) We create a set of messages by producing *transformed copies* of X'. We take a set $\mathbb{T}$ of transformations, where each transformation transforms instances into instances. Afterwards, we draw m transformations $\tau_1, \tau_2, \ldots, \tau_m$ from $\mathbb{T}$ uniformly at random. In this way, we can create a set $\mathcal{M} = \{\tau_1 \circ X', \tau_2 \circ X', \ldots, \tau_m \circ X'\}$ of messages that are analogous to the set of messages that we created in the ideal variant. We impose the following requirement on $\mathbb{T}$ so that $\mathcal{M}$ looks like the set $\mathcal{M}$ that we would obtain in the ideal variant.

- $\sum_\tau p(c \mid \tau \circ X') \in \left[\frac{|\mathbb{T}|}{|\mathcal{C}|}(1-\rho), \frac{|\mathbb{T}|}{|\mathcal{C}|}(1+\rho)\right]$, for some small $\rho > 0$. The reason for this assumption is that we we do not want the probability mass of all these codewords to be concentrated in a narrow subset of $\mathcal{C}$, as this reduces the number of different messages that the sender can communicate to the receiver. This can be ensured that, for each $c \in \mathcal{C}$, the total probability mass c gets is $\sum_\tau p(c \mid \tau \circ X') \approx \frac{|\mathbb{T}|}{|\mathcal{C}|}$.
- The transformations do not alter an instance's randomness.

Observe that such a set of transformations does not necessarily always exist. For example, if the algorithm under evaluation always produces the same constant distribution $p(\cdot \mid X')$, independent of X', then no set of transformations will be able to achieve the requirement above. In this case, however, there is no need to build a channel to evaluate such an algorithm, as it is clear that the algorithm is not producing any value from the data.

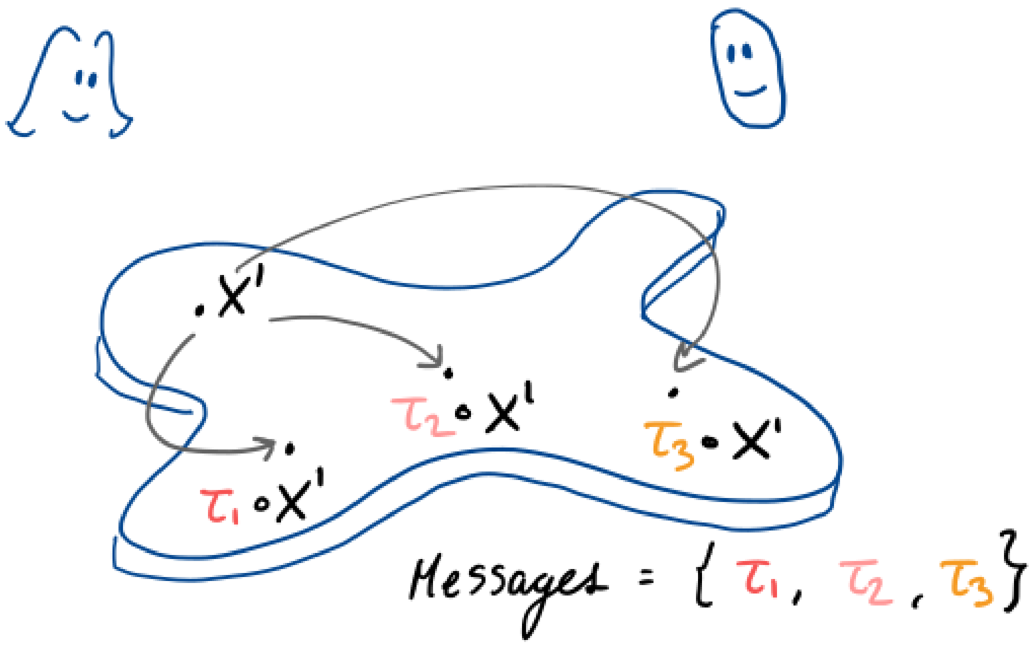


Figure 8

Code (Figure 9) The encoding and decoding functions are the analogous of the ideal variant. The codeword of a message $\tau \in \mathcal{M}$ is $p(\cdot \mid \tau \circ X')$. When given a codeword $p(\cdot \mid Y)$, the decoding function outputs the message τ for which $k(Y, \tau \circ X') \geq k(Y, \sigma \circ X')$, for every $\sigma \in \mathcal{M}$.

Channel (Figure 10) When the sender inputs $p(\cdot \mid \tau \circ X')$ to the channel, the channel outputs $p(\cdot \mid \tau \circ X'')$, as in the ideal variant.

Protocol The sender and the receiver agree on a set $\mathbb{T}$ of transformations and use the algorithm under evaluation $\mathcal{A}$ to compute the code's encoding and decoding functions. A set $\mathcal{M} = \{\tau_1, \ldots, \tau_m\}$ of m messages is drawn uniformly at random from $\mathbb{T}$. A message $\tau \in \mathcal{M}$ is selected uniformly at random and without the receiver's knowledge. The sender then sends the codeword $p(\cdot \mid \tau \circ X')$ through the channel. The receiver gets $p(\cdot \mid \tau \circ X'')$ and uses the decoding function to guess which message the sender sent (Figure 6). The receiver succeeds by correctly guessing τ.

5.4 Probability of a communication error

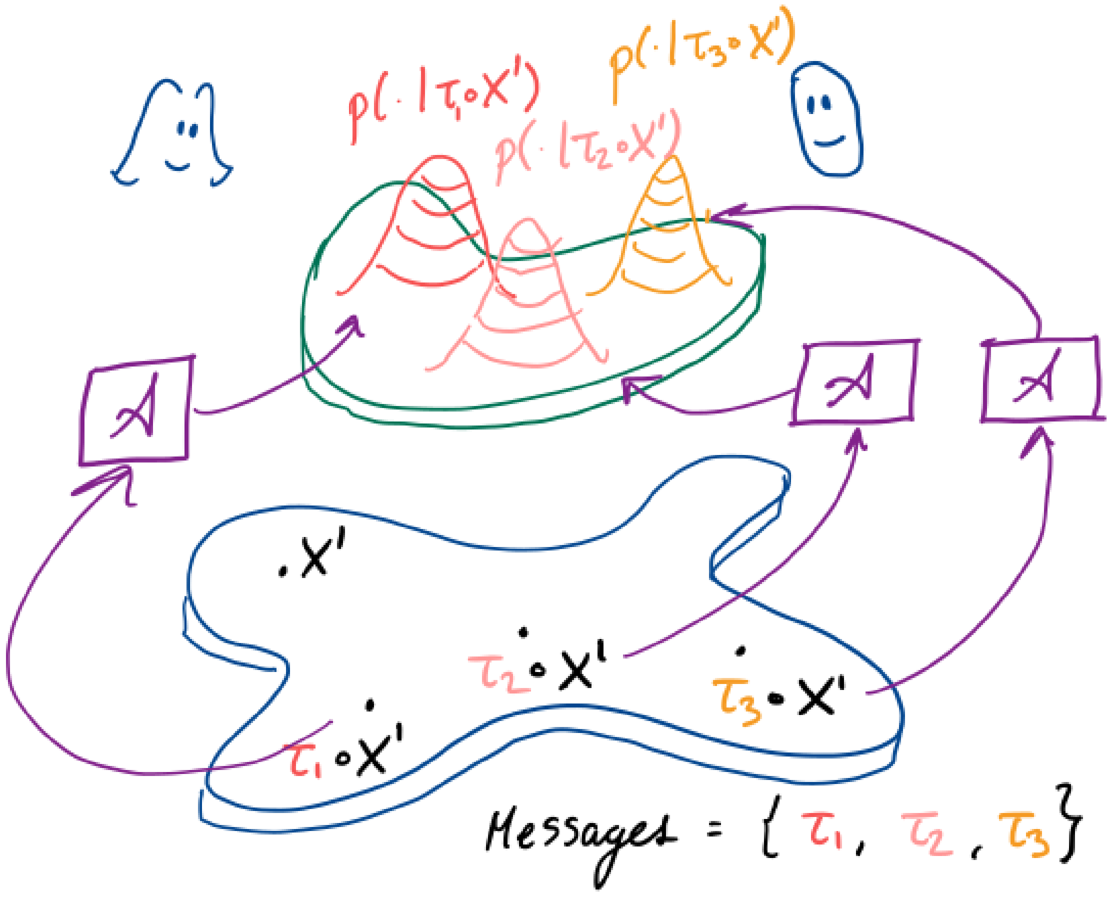
Theorem 6. The probability of a communication error is bounded above by

$$P_{(n)} = \exp\left(-I + \log m + \epsilon \log |\mathcal{C}|\right), \tag{5}$$

where

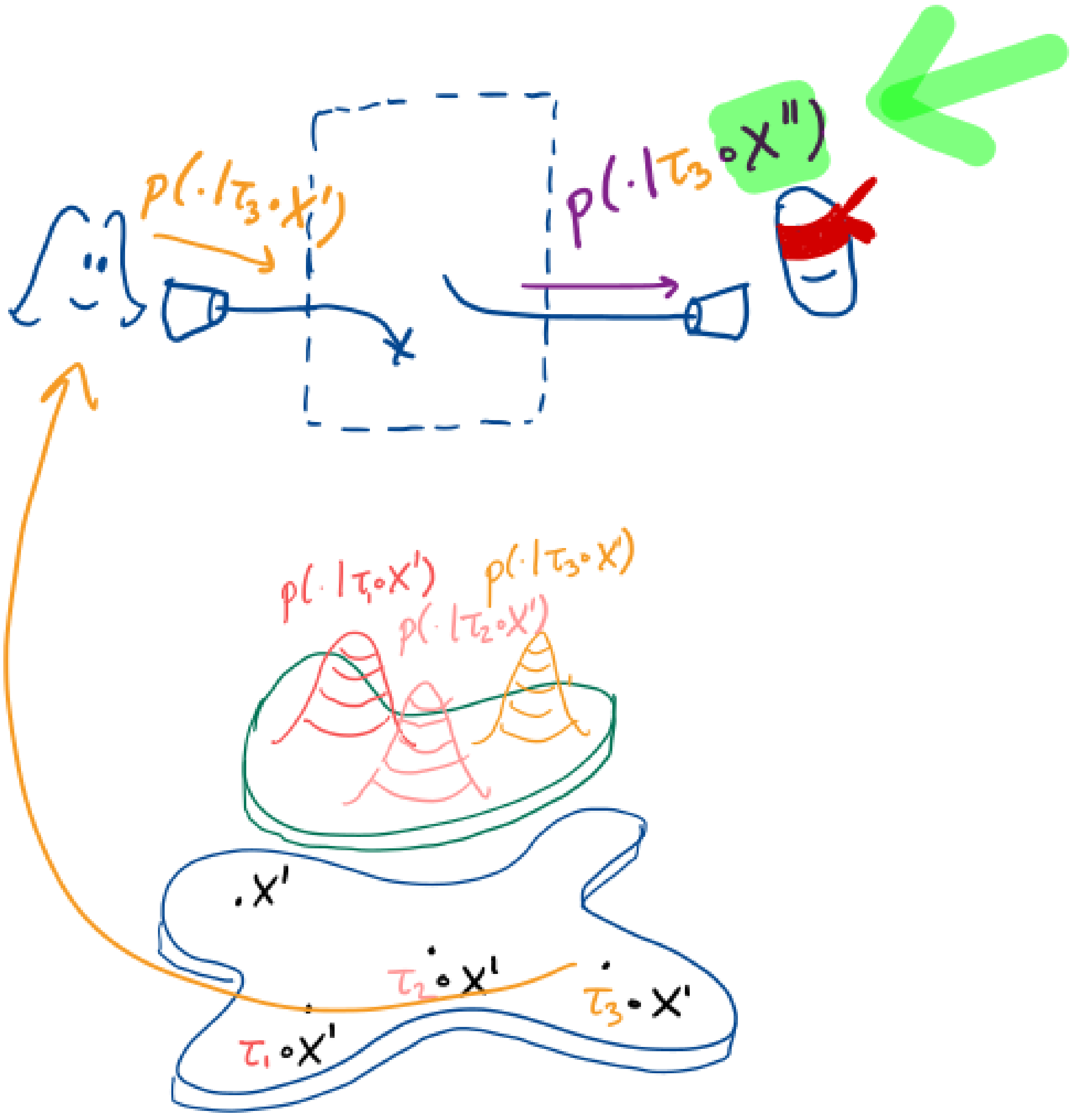
$$I := \mathbb{E}_{X',X''}\left[\log\left(|\mathcal{C}|\, k(X', X'')\right)\right] \tag{6}$$

One can show that $P_{(n)} \to 0$ as $n \to \infty$ by choosing ϵ sufficiently small and ensuring that $I - \log m = \Omega(n)$.



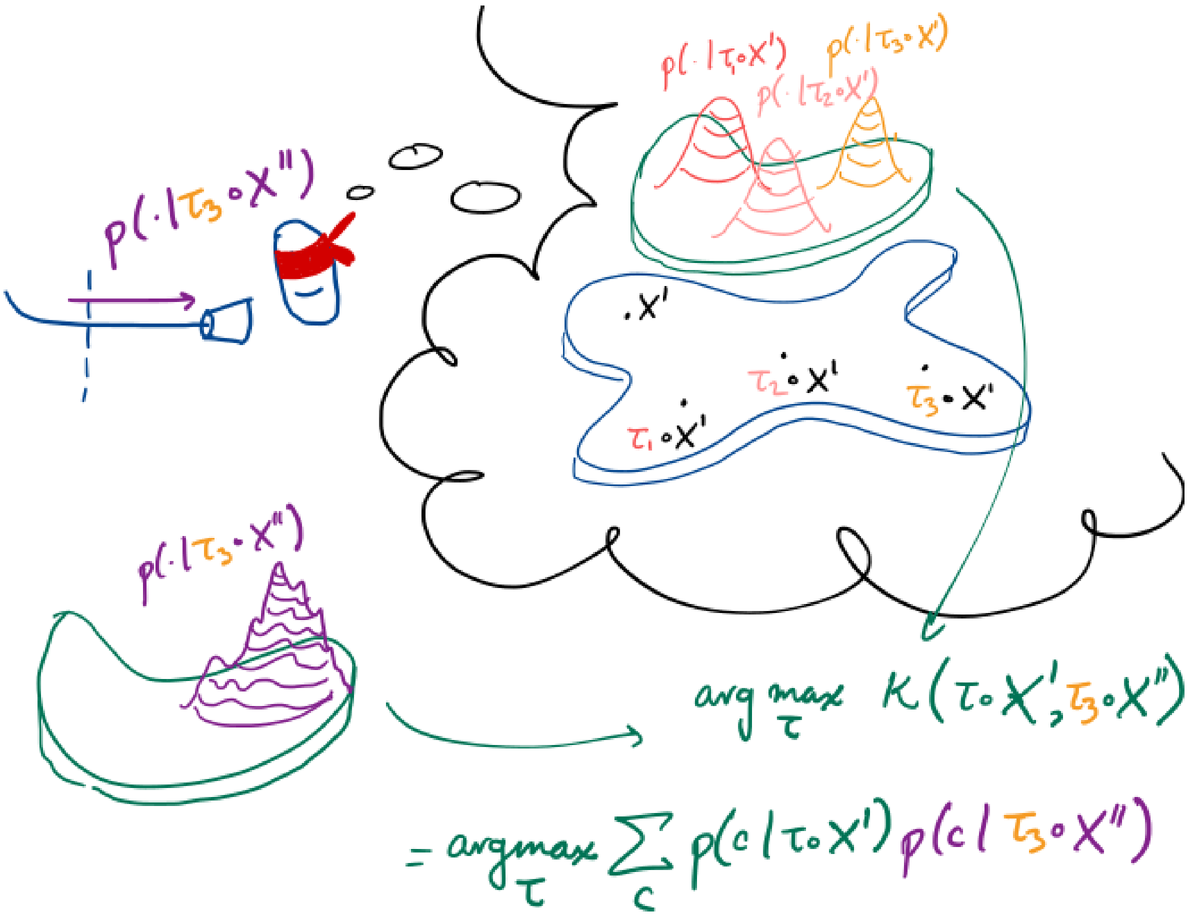
$p(\cdot \mid \tau_1 \circ X')$
$p(\cdot \mid \tau_2 \circ X')$
$p(\cdot \mid \tau_3 \circ X')$
$\mathcal{A}$
$\mathcal{A}$
$\mathcal{A}$
$\cdot X'$
$\tau_1 \circ X'$
$\tau_2 \circ X'$
$\tau_3 \circ X'$
Messages $= \{\tau_1, \tau_2, \tau_3\}$

Figure 9



$p(\cdot|\tau_3\circ X')$
$p(\cdot|\tau_3\circ X'')$
$p(\cdot|\tau_1\circ X')$
$p(\cdot|\tau_3\circ X')$
$p(\cdot|\tau_2\circ X')$
$\cdot X'$
$\tau_2\circ X'$
$\tau_3\circ X'$
$\tau_1\circ X'$

Figure 10



$p(\cdot|\tau_1 \circ X')$
$p(\cdot|\tau_3 \circ X')$
$p(\cdot|\tau_2 \circ X')$
$p(\cdot|\tau_3 \circ X'')$
$\cdot X'$
$\tau_2 \circ X'$
$\tau_3 \circ X'$
$\tau_1 \circ X'$
$p(\cdot|\tau_3 \circ X'')$
$\arg\max_\tau K(\tau \circ X', \tau_3 \circ X'')$
$= \arg\max_\tau \sum_c p(c|\tau \circ X') p(c|\tau_3 \circ X'')$

Figure 11

Proof. Let τ_s be the message sent by the sender and let $\hat{\tau}$ be the message guessed by the receiver. Just as in the proof of Theorem 5, one can show that $\mathbb{P}(\hat{\tau} \neq \tau_s) = \mathbb{P}(\hat{\tau} \neq id)$, where id is some arbitrary transformation. Without loss of generality, we assume that id is the identity transformation.

Using the definition of communication error, we get that

$$\mathbb{P}(\hat{\tau} \neq id) = \mathbb{P}\left(\max_{\tau \neq id} \kappa(\tau \circ X', X'') \geq \kappa(X', X'') \mid id\right).$$

Applying the union bound, we get

$$\mathbb{P}(\hat{\tau} \neq id) \leq \sum_{\tau \neq id} \mathbb{P}(\kappa(\tau \circ X', X'') \geq \kappa(X', X'') \mid id).$$

We now rewrite probabilities as expectations

$$\begin{aligned}
\mathbb{P}(\hat{\tau} \neq id) &\leq \sum_{\tau \neq id} \mathbb{P}(\kappa(\tau \circ X', X'') \geq \kappa(X', X'') \mid id) \\
&= \sum_{\tau \neq id} \mathbb{E}_{X', X''}\left[\mathbb{E}_\tau\left[\mathbb{I}\left\{\kappa(\tau \circ X', X'') \geq \kappa(X', X'')\right\} \mid id, X', X''\right]\right] \\
&= \sum_{\tau \neq id} \mathbb{E}_{X', X''}\left[\mathbb{P}(\kappa(\tau \circ X', X'') \geq \kappa(X', X'') \mid id, X', X'')\right].
\end{aligned}$$

Here, $\mathbb{I}$ is the indicator function.

We now apply Markov's inequality.

$$\begin{aligned}
\mathbb{P}(\hat{\tau} \neq id) &\leq \sum_{\tau \neq id} \mathbb{E}_{X', X''}\left[\mathbb{P}(\mathbb{I}\left\{\kappa(\tau \circ X', X'') \geq \kappa(X', X'')\right\} \mid id, X', X'')\right] \\
&\leq \sum_{\tau \neq id} \mathbb{E}_{X', X''}\left[\frac{\mathbb{E}_\tau\left[\kappa(\tau \circ X', X'') \mid id, X', X''\right]}{\kappa(X', X'')}\right].
\end{aligned}$$

We now compute an upper bound for the numerator.

$$\begin{aligned}
\mathbb{E}_\tau\left[\kappa(\tau \circ X', X'') \mid X', X''\right] &= \mathbb{E}_\tau\left[\sum_c p(c \mid \tau \circ X') p(c \mid X'') \mid X', X''\right] \\
&= \sum_c p(c \mid X'') \mathbb{E}_\tau\left[p(c \mid \tau \circ X')\right] \\
&= \sum_c p(c \mid X'') \sum_\tau \frac{1}{|\mathbb{T}|} p(c \mid \tau \circ X') \\
&\leq \frac{1}{|\mathbb{T}|} \frac{|\mathbb{T}|}{|\mathcal{C}|}(1+\rho) \sum_c p(c \mid X'') = (1+\rho)\frac{1}{|\mathcal{C}|}.
\end{aligned}$$

For the last inequality, we used the assumption that $\sum_\tau p(c \mid \tau \circ X') \in \left[\frac{|\mathbb{T}|}{|\mathcal{C}|}(1-\rho), \frac{|\mathbb{T}|}{|\mathcal{C}|}(1+\rho)\right]$.

We have then that

$$\begin{aligned}
\mathbb{P}\left(\hat{\tau} \neq id\right) &\leq \sum_{\tau \neq id} \mathbb{E}_{X',X''}\left[\frac{1+\rho}{|\mathcal{C}|\,\kappa\left(X',X''\right)}\right] \\
&= (1+\rho)\,(m-1)\,\mathbb{E}_{X',X''}\exp\left(-\log\left(|\mathcal{C}|\,\kappa\left(X',X''\right)\right)\right) \\
&\leq m(1+\rho)\;\mathbb{E}_{X',X''}\exp\left(-\log\left(|\mathcal{C}|\,\kappa\left(X',X''\right)\right)\right) \\
&= (1+\rho)\mathbb{E}_{X',X''}\exp\left(-\log\left(|\mathcal{C}|\,\kappa\left(X',X''\right)\right)+\log m\right) \\
&\approx\; \mathbb{E}_{X',X''}\exp\left(-\hat{I}+\log m\right).
\end{aligned}$$

Here, for simplicity, we assumed $1+\rho \approx 1$.

To provide an upper bound to $\mathbb{P}\left(\hat{\tau} \neq id\right)$, we must bound the behavior of the random variable $\hat{I} = \log\left(|\mathcal{C}|\,\kappa\left(X',X''\right)\right)$. To do this, we assume that $\hat{I}$ satisfies *an asymptotic equipartition property* in the sense that, as $n \to \infty$, $\hat{I} \to I$, where $I = \mathbb{E}_{X',X''}\log\left(|\mathcal{C}|\,\kappa\left(X',X''\right)\right)$, the expected log posterior agreement. Under this assumption, for every $\epsilon,\delta > 0$, there is $n_0 \in \mathbb{N}$ such that for any $n > n_0$,

$$\mathbb{P}\left(\left|\hat{I}-I\right| \leq \epsilon \log|\mathcal{C}|\right) > 1-\delta.$$

With this assumption, we can derive the following:

$$\begin{aligned}
\mathbb{P}\left(\hat{\tau} \neq id\right) &\leq \mathbb{E}_{X',X''}\exp\left(-\hat{I}+\log m\right) \\
&= \mathbb{E}_{X',X''}\exp\left(-\hat{I}+(I-I)+\log m\right) \\
&= \mathbb{E}_{X',X''}\exp\left(-I+(I-\hat{I})+\log m\right) \\
&\leq \mathbb{E}_{X',X''}\exp\left(-I+\left|I-\hat{I}\right|+\log m\right) \\
&\leq \mathbb{E}_{X',X''}\exp\left(-I+\epsilon\log|\mathcal{C}|+\log m\right) \\
&= \exp\left(-I+\epsilon\log|\mathcal{C}|+\log m\right).
\end{aligned}$$

This concludes the proof. □

Recall that the goal is to be able to maximize the number of distinguishable messages that can be sent through the channel. Hence, we must aim to make both m and I as large as possible. The algorithm can only influence I and, therefore, good algorithms *shall maximize the expected log posterior agreement.*

Computing I requires the underlying distribution of X' and X'', which we assume to be unknown. In this case, we can approximate I with *the empirical log posterior agreement*

$$\frac{1}{L}\sum_{\ell \leq L}\left[\log\left(|\mathcal{C}|\,k(X'_\ell,X''_\ell)\right)\right], \tag{7}$$

where $\{X'_1, X''_1, \ldots, X'_L, X''_L\}$ is a set of observations.

Finally, we remark some analogies with Shannon's channel coding theorem. The quantity $\frac{1}{n}\mathbb{E}_{X',X''}\log\left(|\mathcal{C}|\,\kappa\left(X',X''\right)\right)$ plays the role the input-output mutual information. The value $\log m/n$ plays the role of the code rate.