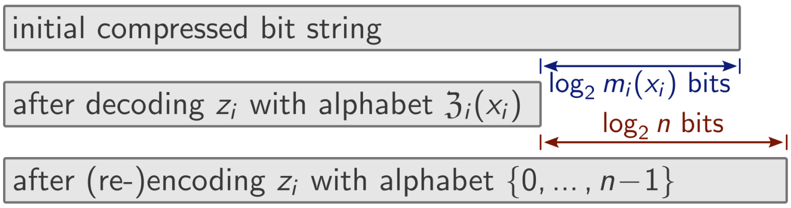
# Bits-Back Coding And Variational Inference

Lecture 7 (3 June 2022); lecturer: Robert Bamler
more course materials online at https://robamler.github.io/teaching/compress22/

## Recap from last lecture: Asymmetric Numeral Systems (ANS)

- stream code that operates as a stack (i.e., "last in first out")

- uses "bits-back" trick (see illustration)



initial compressed bit string

after decoding $z_i$ with alphabet $\mathfrak{Z}_i(x_i)$

$\log_2 m_i(x_i)$ bits

$\log_2 n$ bits

after (re-)encoding $z_i$ with alphabet $\{0, \ldots, n-1\}$

## Today: generalize bits-back trick to arbitrary latent variable models

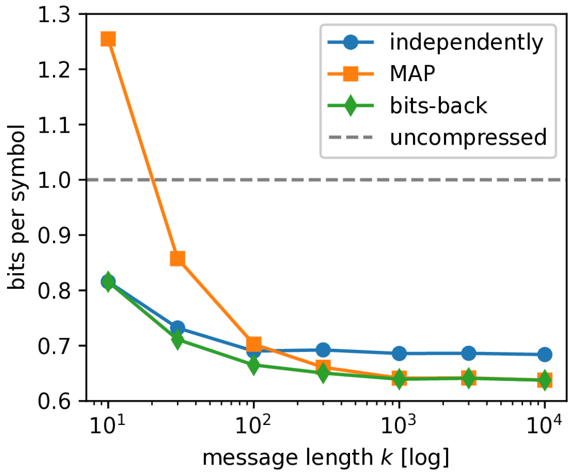
- This will not only generalize the main trick that's used in ANS, it will also use ANS internally.

- We'll also see that an important method from probabilistic machine learning, variational inference, follows directly from the bits-back coding objective. We'll use variational inference in the next lecture to introduce an important class of deep probabilistic models, so-called variational autoencoders.

## Spoiler: Main Results (see Problem Set 7)

Today, we'll
- (re-)introduce latent variable models
- think about three differnent ways how we could use latent variable models for data compression, starting from a very simple and naive method and culminating in the so-called bits-back coding algorithm.

On the problem set, you will
- focus on a concrete toy example latent variable model;
- implement of all three compression methods with latent variable models that we'll discuss today;
- compare the performance of these methods, culminating in the plot on the right.



## Reminder: Latent Variable Models

**Example: Consider the following hypothetical news headlines:**

"Parliament Votes on New Labor Bill."

"Labor Union Votes to Extend Strikes."

topic "politics"

"Soccer Player Scores First Goal Since Joining New Team."

"Guest Team is Leading by One Goal."

topic "sports"

⋮

**Observation:** certain words tend to appear together ("labor" and "votes", "goal" and "team")

⇒ words are correlated: $P(X_i, X_j) \neq P(X_i)\,P(X_j)$

$X_i$

$i, j$: positions within headline

**Possible Explanation:**

- each headline corresponds to a some "topic"
- depending on the topic, certain words are more frequent

latent variable $z$ (not part of the message but helpful for modeling the generative process)

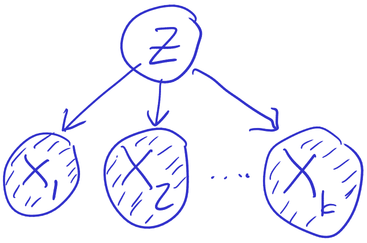
**Latent variable model:**

- message (newspaper headline): sequence of words $\underline{x} = (x_1, x_2, \ldots, x_k)$
- latent topic z is shared across these words

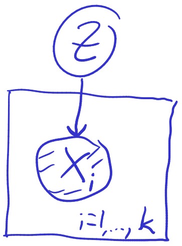
⇒ joint probability distribution:

$$P(\underline{X}, Z) = \underbrace{P(Z)}_{\text{"prior": distribution of topics}} \prod_{i=1}^{k} \underbrace{P(X_i|Z)}_{\text{"likelihood": word frequencies within a given topic}}$$

pictorially:



shorthand:



Note: despite its simplicity, this kind of so-called "topic model" is actually very powerful and widely used in research and industry for unsupervised categorization of large amounts of texts (e.g., websites, newspaper articles, patents, ...). If you're curious, look up "Latent Dirichlet Allocation" (LDA), first introduced by Pritchard et al. (2000) in the context of genetics, and later popularized in the natural language ML community by Blei & Ng (2003).

Recall: we only want to encode the message $\underline{x}$, not the latent variable.

⇒ marginal probability distribution of the message:

$$P(\underline{X}) = \sum_{z} P(\underline{X}, Z=z) = \sum_{z} \left( P(Z=z) \prod_{i=1}^{k} P(X_i|Z=z) \right)$$

recall: this kind of marginal distribution can indeed capture correlations between symbols (here: words), as we showed in Problem 5.2 (d).

Problem: P($\underline{X}$) is a complicated probability distribution; we can't easily write it in an autoregressive way.
⇒ Not obvious how we can use P(X) for compression.

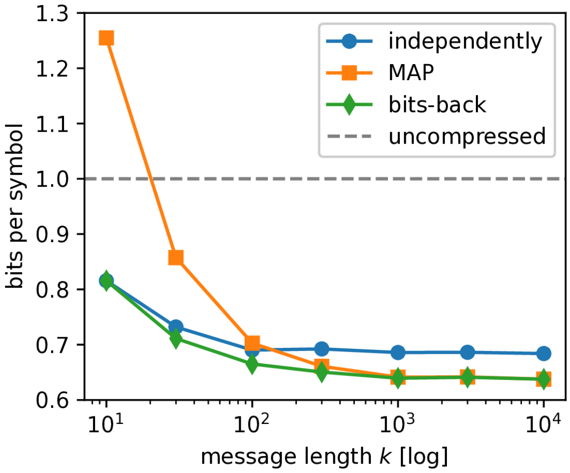
$P(\underline{X}) = P(X_1)\,P(X_2|X_1)\,P(X_3|X_1,X_2)\ldots$

(always possible in theory but prohibitively computationally expensive.

# Data Compression With Latent Variable Models

**Problem set (strongly recommended! feel free to work in teams):**

implement & compare 3 compression methods for latent variable models:



- Problem 7.2: naive method: ignore correlations and treat words as independent

$$P'(\underline{X}) = \prod_{i=1}^{k} P'(X_i)$$

$$\Rightarrow \mathbb{E}_p[R(\underline{X})] \underset{\uparrow \text{ Problem 5.2 (a)}}{=} k\, H_p[X_i] \geq H[\underline{X}]$$

- Problem 7.3: "MAP estimate method": encode some dummy z*, then encode each symbol $X_i$ using the likelihood P($X_i$| Z=z*)

$$\Rightarrow R(\underline{X}) = -\log_2 P(Z=z^*) - \sum_{i=1}^{k} \log_2 P(X_i=x_i \mid Z=z^*)$$

- Problem 7.4: bits-back coding $\Rightarrow \mathbb{E}_p[R_{net}(\underline{X})] = H_p(\underline{X})$, i.e., optimal

Thus, the bits-back coding algorithm, which we'll discuss below, has the same (net) bit rate as if we were using the marginal distribution P(X) for compression, even though we won't directly use this marginal distribution.

## MAP Estimate Method

Idea: - find some dummy z*, then encode each symbol $X_i$ using the likelihood P(X | Z=z*);
- use some (near) optimal stream code, like range coding or ANS.

$$\Rightarrow R(\underline{X}) = -\log_2 P(Z=z^*) - \sum_{i=1}^{k} \log_2 P(X_i=x_i \mid Z=z^*) = -\log_2 P(\underline{X}=\underline{x}, Z=z^*)$$

Question: which value of z* should we chose

↳ we don't care what the "true" value of Z is (if there even is one), we just want to transmit $\underline{x}$ in as few bits as possible

$$\Rightarrow \text{let } z^* := \arg\min_z \left[-\log_2 P(\underline{X}=\underline{x}, Z=z)\right]$$

$$= \arg\max_z P(\underline{X}=\underline{x}, Z=z)$$

$$\left(= \arg\max_z \underbrace{P(Z=z \mid \underline{X}=\underline{x})}_{\text{"posterior"} = \frac{P(\underline{X}=\underline{x}, Z=z)}{P(\underline{X}=\underline{x})}} = \underbrace{\text{"maximum a-posteriori"}}_{\text{MAP}}\right)$$

### Encoding scheme with MAP-estimate method:

1) find z* as described above
2) encode z* using the prior model P(Z) and each symbol $x_i$ using the likelihood model $P(X_i \mid Z=z^*)$

   [if using a range coder, encode z* first and then the symbols $x_i$ ; if using ANS, encode first the symbols $x_i$ and then z* so that the decoder can decode z* first]

### Decoding scheme with MAP-estimate method:

1) decode z* using the prior model P(Z)
2) decode all symbols $x_i$ using the likelihood model $P(X_i \mid Z=z^*)$
3) throw away z* → overhead

### Bit rate overhead of MAP-estimate method:

$$\underbrace{-\log_2 P(\underline{X}=\underline{x}, Z=z^*)}_{\text{actual bit rate}} - \underbrace{\left(-\log_2 P(\underline{X}=\underline{x})\right)}_{\substack{\text{information content of message}\\ =\text{optimal bit rate}}} = -\log_2 \frac{P(\underline{X}=\underline{x}, Z=z^*)}{P(Z=z^*)} = \underbrace{-\log_2 P(Z=z^* \mid \underline{X}=\underline{x})}_{\substack{\text{information content}\\ \text{of the MAP estimate } z^*\\ \text{under the \underline{posterior}}\\ \text{distribution}}}$$

**Understanding the overhead:** recall our news headlines:

"Parliament Votes on New Labor Bill."

"Labor Union Votes to Extend Strikes."

} topic "politics"

"Soccer Player Scores First Goal Since Joining New Team."

"Guest Team is Leading by One Goal."

} topic "sports"

⋮

Now consider the following hypothetical headline:

"Parliament Votes on Aid for Community Sports Teams."

→ Could be encoded either with z* = "politics" or with z* = "sports"

⇒ 2 different compressed representations that encode the same message → wasteful

This uncertainty about the latent variable Z even when we know the message $\underline{x}$ is exactly what is described by the posterior distribution $P(Z \mid \underline{X}=\underline{x})$.

recall: overhead $= \underbrace{-\log_2 P(Z=z^* \mid \underline{X}=\underline{x})}_{\text{high posterior uncertainty}}$

⇔ low maximum posterior probability $P(Z=z^* \mid \underline{X}=\underline{x})$

⇔ high overhead

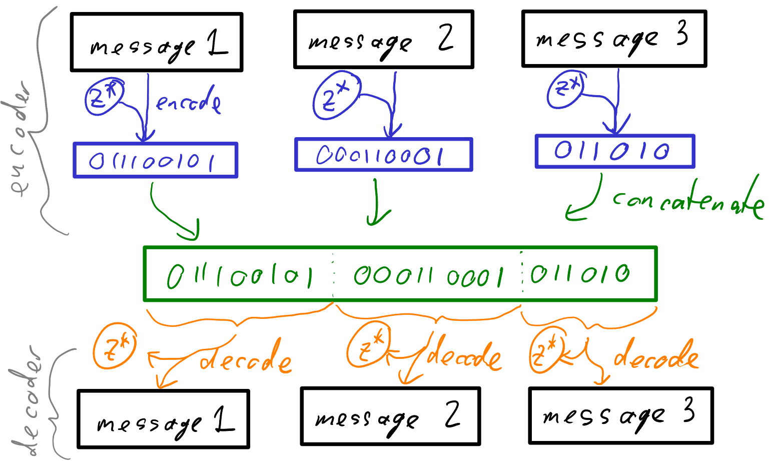
# Bits-Back Coding

[Wallace 1990, Hinton & Camp 1993]
practical: BB-ANS [Townsend et al. 2019]
lossy: [Yang, RB, Mandt, 2020]

Idea: "piggyback" some additional side information into the choice of z*

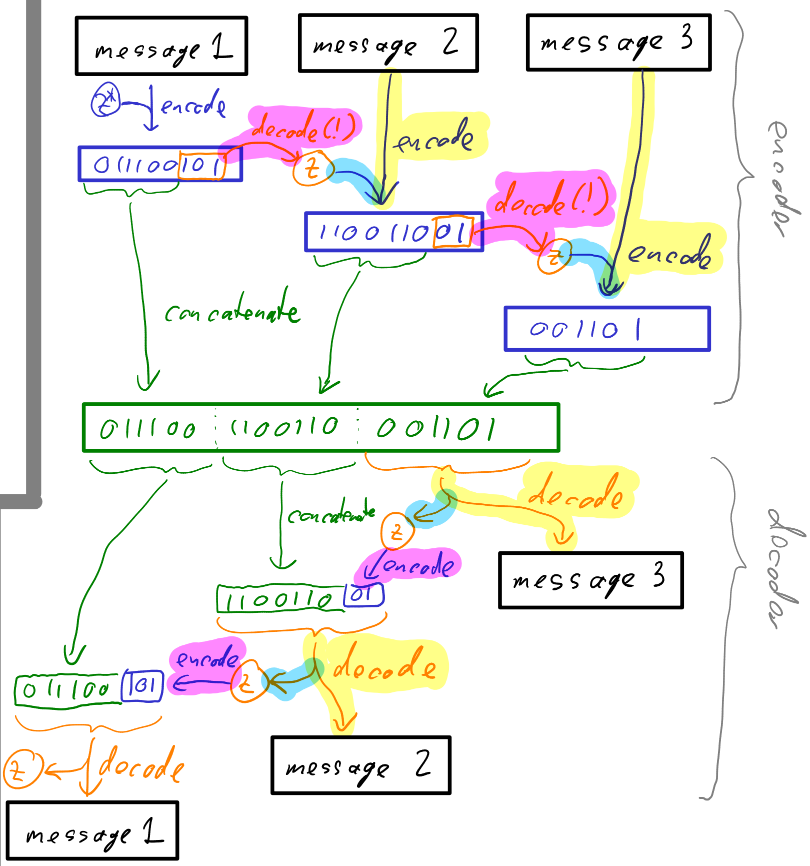
Typical setup:
- communicate multiple messages (e.g., multiple image pages or multiple frames of a video) over a single channel
- side information = any previously encoded data

## MAP-estimated method (no bits-back coding):



## Bits-back coding:

operates as a stack (i.e., "last in first out")



## Algorithm (Bits-Back Coding):

bit string

subroutine bb_encode($\underline{x}$, existing_compressed, P)

```
• $z$ ← decode with ANS from existing_compressed using $P(Z|\underline{X}=\underline{x})$
• encode $\underline{x}$ onto existing_compressed using ANS & model $P(\underline{X}|Z=z)$
• encode $z$ onto existing_compressed using ANS & prior model $P(Z)$
• return existing_compressed
```

subroutine bb_decode(compressed, P):

```
• $z$ ← decode with ANS from compresse using prior model $P(Z)$
• $\underline{x}$ ← decode with ANS from compressed using likelihood $P(\underline{X}|Z=z)$
• encode $z$ onto compressed using posterior $P(Z|\underline{X}=\underline{x})$
• return ($\underline{x}$, compressed)
```

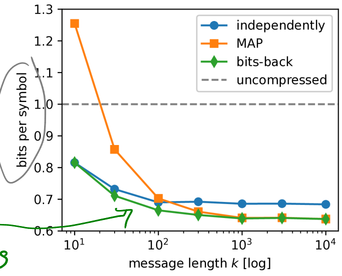
## Net bit rate of bits-back coding:

$$\left.\begin{aligned} R_{net}(\underline{x}) = & -\log_2 P(\underline{X}=\underline{x}|Z=z) \\ & -\log_2 P(Z=z) \\ & -(-\log_2 P(Z=z|\underline{X}=\underline{x})) \end{aligned}\right\} \textcircled{*}$$

$$= -\log_2 \frac{\cancel{P(\underline{X}=\underline{x}, Z=z)}\ \cancel{P(Z=z)}\ P(\underline{X}=\underline{x})}{\cancel{P(Z=z)}\ \cancel{P(\underline{X}=\underline{x}, Z=z)}}$$

$$= -\log_2 P(\underline{X}=\underline{x}) \rightarrow \text{optimal !}$$

actual(!) measured bit rates (not just estimates)



bits-back coding has lowest bit rate ∀ message lengths

→ Problem set:

**Note:** Recall that we already used the bits-back trick inside the ANS algorithm itself (last lecture). There, bits-back coding was a bit simpler because the prior was a uniform distribution and the likelihood was deterministic. Can you identify the steps of the general bits-back algorithm in the special example of ANS?

Hint: the yellow step in the encoder and decoder were not necessary for ANS. Can you explain why? How many bits would encoding $\underline{x}$ with the likelihood model $P(\underline{X} \mid Z=z)$ contribute when the likelihood is deterministic?

## Variational Inference (Teaser)

- The above derivation that the net bit rate of bits back coding is the optimal bit rate only works if we use the posterior distribution P(Z | X=x) for decoding z in bb_encode.

- Since we cannot outperform the optimal bit rate (in expectation), using any other distribution Q(Z) instead of the posterior would lead to a higher bit rate.

- Problem: the true posterior is usually hard to calculate:

$$P(z \mid \underline{X}=\underline{x}) = \frac{P(z)\, P(\underline{X}=\underline{x} \mid z)}{\underbrace{\int P(z=z')\, P(\underline{X}=\underline{x} \mid z=z')\, dz'}_{=P(\underline{X}=\underline{x})}}$$

← computationally intractable integral except in very special models

Idea: instead of the true posterior, use some parameterized candidate distribution $Q_\phi(Z)$, and minimize expression $\circledast$ for the net bit rate over the parameters ϕ (where we replace the posterior in $\circledast$ by $Q_\phi(Z)$).

→ This method is called "Variational Inference", and it is an important method in modern probabilistic machine learning, far beyond applications for data copmression.

→ In the next lecture, we'll apply (a generalization of) Variational Inference to deep latent variable models (i.e., latent variable models that are parameterized by deep neural networks). This will lead to the popular variational autoencoder (VAE) model architecture.