Compressing Multisets with Large Alphabets

Daniel Severo[123] James Townsend[4]
Ashish Khisti[2] Alireza Makhzani[23] Karen Ullrich[1]

[1]Meta AI [2]University of Toronto
[3]Vector Institute for AI [4]University of Amsterdam

Data Compression Conference, 2022

Outline

Problem setting

Problem setting

Given a sequence of i.i.d. symbols $X^n = (X_1, \ldots, X_n)$ with entropy

$$H(X^n) = nH(X) = n\mathsf{E}\left[-\log P_X(X)\right]$$

Problem setting

Given a sequence of i.i.d. symbols $X^n = (X_1, \ldots, X_n)$ with entropy

$$H(X^n) = nH(X) = n\mathsf{E}\left[-\log P_X(X)\right]$$

we want to losslessly compress the multiset

$$\mathcal{M} = f(X^n) = \{X_1, \ldots, X_n\}$$

at rate $H(\mathcal{M}) \leq H(X^n)$.

Motivation

Motivation

How to achieve $H(\mathcal{M}) \leq H(X^n)$?

Motivation

How to achieve $H(\mathcal{M}) \leq H(X^n)$?

– Steinruecken (2016): rate-optimal for any alphabet $\mathcal{A}$

Motivation

How to achieve $H(\mathcal{M}) \leq H(X^n)$?

- Steinruecken (2016): rate-optimal for any alphabet $\mathcal{A}$
  - Compress frequency count of symbols in $\mathcal{M}$ (vector in $\mathsf{N}^{|\mathcal{A}|}$)

Motivation

How to achieve $H(\mathcal{M}) \leq H(X^n)$?

- Steinruecken (2016): rate-optimal for any alphabet $\mathcal{A}$
  - Compress frequency count of symbols in $\mathcal{M}$ (vector in $\mathsf{N}^{|\mathcal{A}|}$)
  - Inefficient when $|\mathcal{A}| \gg n$, requires $\mathcal{O}(|\mathcal{A}|)$ steps

Motivation

How to achieve $H(\mathcal{M}) \leq H(X^n)$?

- Steinruecken (2016): rate-optimal for any alphabet $\mathcal{A}$
  - Compress frequency count of symbols in $\mathcal{M}$ (vector in $\mathsf{N}^{|\mathcal{A}|}$)
  - Inefficient when $|\mathcal{A}| \gg n$, requires $\mathcal{O}(|\mathcal{A}|)$ steps

- Compress X^n instead: efficient, if X_i are i.i.d.

Motivation

How to achieve $H(\mathcal{M}) \leq H(X^n)$?

- Steinruecken (2016): rate-optimal for any alphabet $\mathcal{A}$
  - Compress frequency count of symbols in $\mathcal{M}$ (vector in $\mathbb{N}^{|\mathcal{A}|}$)
  - Inefficient when $|\mathcal{A}| \gg n$, requires $\mathcal{O}(|\mathcal{A}|)$ steps

- Compress X^n instead: efficient, if X_i are i.i.d.
  - Entropy code each X_i with $P_X(X_i)$, requires $\mathcal{O}(n)$ steps

Motivation

How to achieve $H(\mathcal{M}) \leq H(X^n)$?

- Steinruecken (2016): rate-optimal for any alphabet $\mathcal{A}$
  - Compress frequency count of symbols in $\mathcal{M}$ (vector in $\mathsf{N}^{|\mathcal{A}|}$)
  - Inefficient when $|\mathcal{A}| \gg n$, requires $\mathcal{O}(|\mathcal{A}|)$ steps

- Compress X^n instead: efficient, if X_i are i.i.d.
  - Entropy code each X_i with $P_X(X_i)$, requires $\mathcal{O}(n)$ steps
  - Sub-optimal, achieves $H(X^n) \geq H(\mathcal{M})$

Motivation

How to achieve $H(\mathcal{M}) \leq H(X^n)$?

- Steinruecken (2016): rate-optimal for any alphabet $\mathcal{A}$
  - Compress frequency count of symbols in $\mathcal{M}$ (vector in $\mathsf{N}^{|\mathcal{A}|}$)
  - Inefficient when $|\mathcal{A}| \gg n$, requires $\mathcal{O}(|\mathcal{A}|)$ steps

- Compress X^n instead: efficient, if X_i are i.i.d.
  - Entropy code each X_i with $P_X(X_i)$, requires $\mathcal{O}(n)$ steps
  - Sub-optimal, achieves $H(X^n) \geq H(\mathcal{M})$

Would like efficient, rate-optimal method for any $\mathcal{A}, n$.

Background

Background: Asymmetric Numeral Systems (ANS)

ANS (Duda, 2009) is an alternative to Arithmetic Coding (AC).

Background: Asymmetric Numeral Systems (ANS)

ANS (Duda, 2009) is an alternative to Arithmetic Coding (AC).

To encode $X = x$ with P_X and CDF F_X,

	AC	ANS
statistics	range $[F_X(x), F_X(x) + P_X(x))$	

Background: Asymmetric Numeral Systems (ANS)

ANS (Duda, 2009) is an alternative to Arithmetic Coding (AC).

To encode $X = x$ with P_X and CDF F_X,

	AC	ANS
statistics	range $[F_X(x), F_X(x) + P_X(x))$	
state	fraction 0.1001	integer 1001

Background: Asymmetric Numeral Systems (ANS)

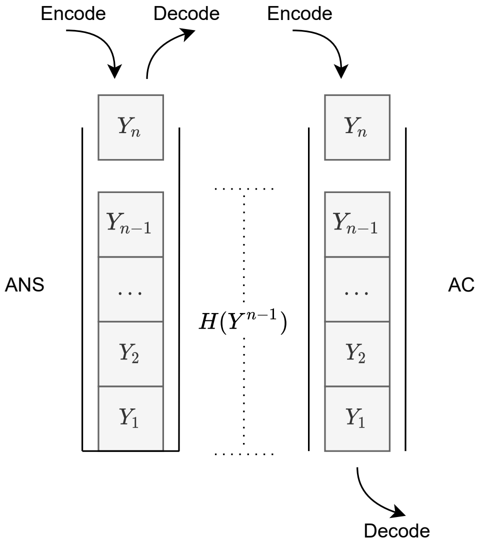
ANS (Duda, 2009) is an alternative to Arithmetic Coding (AC).

To encode $X = x$ with P_X and CDF F_X,

	AC	ANS
statistics	range $[F_X(x), F_X(x) + P_X(x))$	
state	fraction 0.1001	integer 1001
order	queue-like	stack-like

Background: Asymmetric Numeral Systems (ANS)

Key difference: ANS decodes in reverse order



Background: Bits-back with ANS (BB-ANS)

Problem: Given $X = f(Y)$, encode X at rate $R_X = H(X)$

Background: Bits-back with ANS (BB-ANS)

Problem: Given $X = f(Y)$, encode X at rate $R_X = H(X)$ using

- code for Y at rate $H(Y)$

Background: Bits-back with ANS (BB-ANS)

Problem: Given $X = f(Y)$, encode X at rate $R_X = H(X)$ using

- code for Y at rate $H(Y)$ and
- code for $Y \mid X$ at rate $H(Y \mid X)$

Background: Bits-back with ANS (BB-ANS)

Problem: Given $X = f(Y)$, encode X at rate $R_X = H(X)$ using

- code for Y at rate $H(Y)$ and
- code for $Y \mid X$ at rate $H(Y \mid X)$

BB-ANS (Townsend, 2019) achieves $H(X)$ for i.i.d. $X_1, \ldots, X_n$

Background: Bits-back with ANS (BB-ANS)

Problem: Given $X = f(Y)$, encode X at rate $R_X = H(X)$ using

- code for Y at rate $H(Y)$ and
- code for $Y \mid X$ at rate $H(Y \mid X)$

BB-ANS (Townsend, 2019) achieves $H(X)$ for i.i.d. $X_1, \ldots, X_n$

Use ANS stack as a random seed to sample $Y_1 \mid X_1, \ldots, Y_n \mid X_n$

Background: Bits-back with ANS (BB-ANS)

Problem: Given $X = f(Y)$, encode X at rate $R_X = H(X)$ using

- code for Y at rate $H(Y)$ and
- code for $Y \mid X$ at rate $H(Y \mid X)$

BB-ANS (Townsend, 2019) achieves $H(X)$ for i.i.d. $X_1, \ldots, X_n$

Use ANS stack as a random seed to sample $Y_1 \mid X_1, \ldots, Y_n \mid X_n$

Encode $Y_1, \ldots, Y_n$ onto the ANS stack

Background: Bits-back with ANS (BB-ANS)

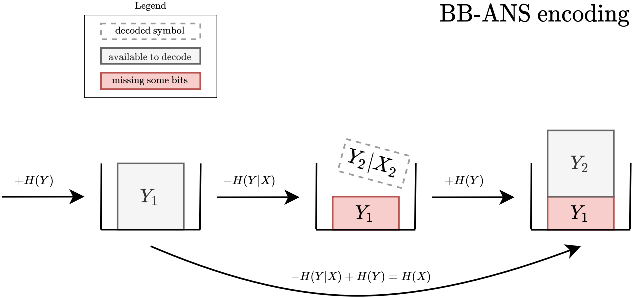
Problem: Given $X = f(Y)$, encode X at rate $R_X = H(X)$ using

- code for Y at rate $H(Y)$ and
- code for $Y \mid X$ at rate $H(Y \mid X)$

BB-ANS (Townsend, 2019) achieves $H(X)$ for i.i.d. $X_1, \ldots, X_n$

Use ANS stack as a random seed to sample $Y_1 \mid X_1, \ldots, Y_n \mid X_n$

Encode $Y_1, \ldots, Y_n$ onto the ANS stack



Background: Bits-back with ANS (BB-ANS)

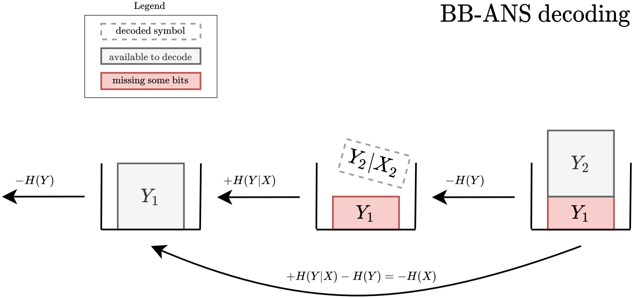
Problem: Given $X = f(Y)$, encode X at rate $R_X = H(X)$ using

- code for Y at rate $H(Y)$ and
- code for $Y \mid X$ at rate $H(Y \mid X)$

BB-ANS (Townsend, 2019) achieves $H(X)$ for i.i.d. $X_1, \ldots, X_n$

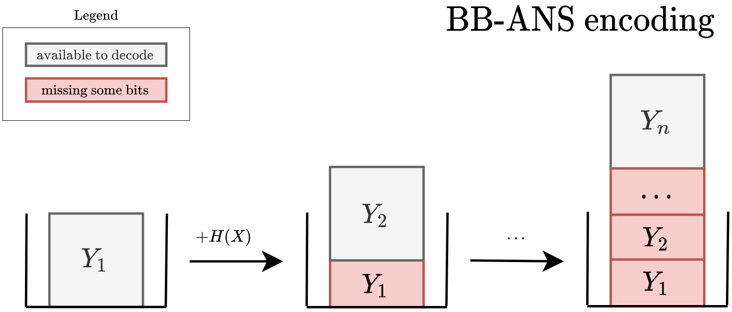
Use ANS stack as a random seed to sample $Y_1 \mid X_1, \ldots, Y_n \mid X_n$

Encode $Y_1, \ldots, Y_n$ onto the ANS stack



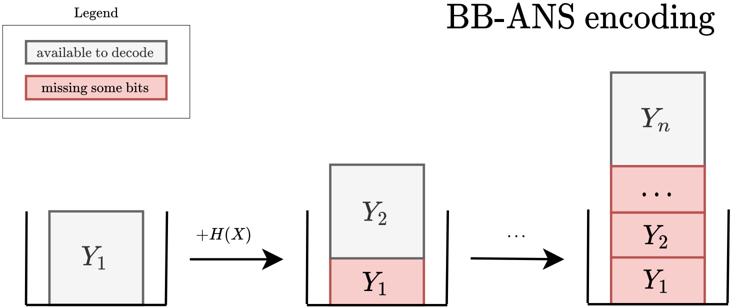
Background: Bits-back with ANS (BB-ANS)

The full picture



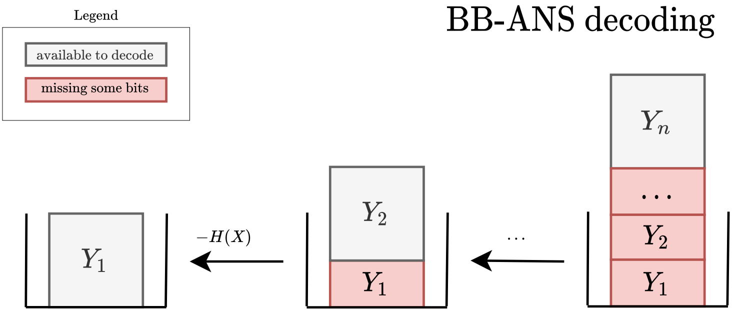
Background: Bits-back with ANS (BB-ANS)

The full picture, with one-time overhead of $+\frac{1}{n}H(Y|X)$



Background: Bits-back with ANS (BB-ANS)

The full picture, with one-time overhead of $+\frac{1}{n}H(Y|X)$



Background: Bits-back with ANS (BB-ANS)

Take-away: BB-ANS gives an operational meaning to the identity

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

where $X = f(Y)$.

Background: Multiset entropy

How does $H(\mathcal{M}) = H(f(X^n))$ relate to $H(X^n)$?

Background: Multiset entropy

How does $H(\mathcal{M}) = H(f(X^n))$ relate to $H(X^n)$?

$$\begin{aligned} H(X^n, \mathcal{M}) &= H(\mathcal{M}) + H(X^n \mid \mathcal{M}) \\ &= H(X^n) + \underbrace{H(\mathcal{M} \mid X^n)}_{=0} \end{aligned}$$

Background: Multiset entropy

How does $H(\mathcal{M}) = H(f(X^n))$ relate to $H(X^n)$?

$$\begin{aligned} H(X^n, \mathcal{M}) &= H(\mathcal{M}) + H(X^n \,|\, \mathcal{M}) \\ &= H(X^n) + \underbrace{H(\mathcal{M} \,|\, X^n)}_{=0} \end{aligned}$$

Multiset entropy

$$H(\mathcal{M}) = H(X^n) - H(X^n \,|\, \mathcal{M})$$

Background: Multiset entropy

How does $H(\mathcal{M}) = H(f(X^n))$ relate to $H(X^n)$?

$$\begin{aligned} H(X^n, \mathcal{M}) &= H(\mathcal{M}) + H(X^n \mid \mathcal{M}) \\ &= H(X^n) + \underbrace{H(\mathcal{M} \mid X^n)}_{=0} \end{aligned}$$

Multiset entropy

$$H(\mathcal{M}) = H(X^n) - H(X^n \mid \mathcal{M})$$

$H(X^n \mid \mathcal{M})$ bits are needed to order symbols in $\mathcal{M}$ to create X^n

Background: Multiset entropy

How does $H(\mathcal{M}) = H(f(X^n))$ relate to $H(X^n)$?

$$\begin{aligned} H(X^n, \mathcal{M}) &= H(\mathcal{M}) + H(X^n \,|\, \mathcal{M}) \\ &= H(X^n) + \underbrace{H(\mathcal{M} \,|\, X^n)}_{=0} \end{aligned}$$

Multiset entropy

$$H(\mathcal{M}) = H(X^n) - H(X^n \,|\, \mathcal{M})$$

$H(X^n \,|\, \mathcal{M})$ bits are needed to order symbols in $\mathcal{M}$ to create X^n

It is often called the "order information"

Method

Method: overview

Recap: BB-ANS gives an operational meaning to the identity

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

where $X = f(Y)$.

Method: overview

Recap: BB-ANS gives an operational meaning to the identity

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

where $X = f(Y)$.

Multiset entropy:

$$H(\mathcal{M}) = H(X^n) - H(X^n \mid \mathcal{M}) = I(\mathcal{M}; X^n).$$

Method: overview

Recap: BB-ANS gives an operational meaning to the identity

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

where $X = f(Y)$.

Multiset entropy:

$$H(\mathcal{M}) = H(X^n) - H(X^n \mid \mathcal{M}) = I(\mathcal{M}; X^n).$$

Naive method: apply BB-ANS for multiset compression

Method: overview

Recap: BB-ANS gives an operational meaning to the identity

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

where $X = f(Y)$.

Multiset entropy:

$$H(\mathcal{M}) = H(X^n) - H(X^n \mid \mathcal{M}) = I(\mathcal{M}; X^n).$$

Naive method: apply BB-ANS for multiset compression

Achieves $H(\mathcal{M})$ on sequence of multisets $\mathcal{M}_1, \mathcal{M}_2, \ldots$

Method: overview

Recap: BB-ANS gives an operational meaning to the identity

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

where $X = f(Y)$.

Multiset entropy:

$$H(\mathcal{M}) = H(X^n) - H(X^n \mid \mathcal{M}) = I(\mathcal{M}; X^n).$$

Naive method: apply BB-ANS for multiset compression

Achieves $H(\mathcal{M})$ on sequence of multisets $\mathcal{M}_1, \mathcal{M}_2, \ldots$

Can we achieve $H(\mathcal{M})$ on a single multiset $\mathcal{M} = f(X^n)$?

Method: overview

Recap: BB-ANS gives an operational meaning to the identity

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

where $X = f(Y)$.

Multiset entropy:

$$H(\mathcal{M}) = H(X^n) - H(X^n \mid \mathcal{M}) = I(\mathcal{M}; X^n).$$

Naive method: apply BB-ANS for multiset compression

Achieves $H(\mathcal{M})$ on sequence of multisets $\mathcal{M}_1, \mathcal{M}_2, \ldots$

Can we achieve $H(\mathcal{M})$ on a single multiset $\mathcal{M} = f(X^n)$?

In other words, can we compress $\mathcal{M}$ to $-\log P_{\mathcal{M}}(\mathcal{M})$ bits?

Method: compressing $\mathcal{M}$ to $-\log P_{\mathcal{M}}(\mathcal{M})$ bits

Construct order information $H(X^n \mid \mathcal{M})$ iteratively by "sampling without replacement" from $\mathcal{M}$. Alternate:

1. Decode sample (w.o. replacement) from $\mathcal{M}$
2. Encode sampled element using P_X

until $\mathcal{M}$ is depleted.

Method: compressing $\mathcal{M}$ to $-\log P_{\mathcal{M}}(\mathcal{M})$ bits

Construct order information $H(X^n \mid \mathcal{M})$ iteratively by "sampling without replacement" from $\mathcal{M}$. Alternate:

1. Decode sample (w.o. replacement) from $\mathcal{M}$
2. Encode sampled element using P_X

until $\mathcal{M}$ is depleted.

$\{\mathtt{a}, \mathtt{b}, \mathtt{b}\}$

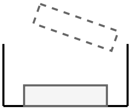
$L(\mathcal{M}) = \varepsilon$

Method: compressing $\mathcal{M}$ to $-\log P_{\mathcal{M}}(\mathcal{M})$ bits

Construct order information $H(X^n \mid \mathcal{M})$ iteratively by "sampling without replacement" from $\mathcal{M}$. Alternate:

1. Decode sample (w.o. replacement) from $\mathcal{M}$
2. Encode sampled element using P_X

until $\mathcal{M}$ is depleted.



$\{\mathtt{a}, \mathtt{b}, \mathtt{b}\}$

$$L(\mathcal{M}) = \varepsilon - \log \frac{1}{2/3}$$

Method: compressing $\mathcal{M}$ to $-\log P_{\mathcal{M}}(\mathcal{M})$ bits

Construct order information $H(X^n \mid \mathcal{M})$ iteratively by "sampling without replacement" from $\mathcal{M}$. Alternate:

1. Decode sample (w.o. replacement) from $\mathcal{M}$
2. Encode sampled element using P_X

until $\mathcal{M}$ is depleted.

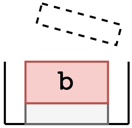
$$L(\mathcal{M}) = \varepsilon - \log \frac{1}{2/3} + \log \frac{1}{P_X(\mathtt{b})}$$

Method: compressing $\mathcal{M}$ to $-\log P_{\mathcal{M}}(\mathcal{M})$ bits

Construct order information $H(X^n \mid \mathcal{M})$ iteratively by “sampling without replacement” from $\mathcal{M}$. Alternate:

1. Decode sample (w.o. replacement) from $\mathcal{M}$
2. Encode sampled element using P_X

until $\mathcal{M}$ is depleted.



$\{\texttt{a}, \texttt{b}\}$

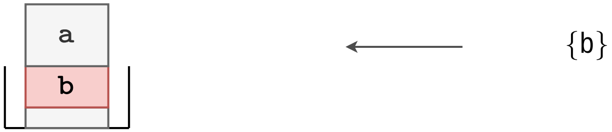
$$L(\mathcal{M}) = \varepsilon - \log \frac{1}{2/3} + \log \frac{1}{P_X(\texttt{b})} - \log \frac{1}{1/2}$$

Method: compressing $\mathcal{M}$ to $-\log P_{\mathcal{M}}(\mathcal{M})$ bits

Construct order information $H(X^n \mid \mathcal{M})$ iteratively by "sampling without replacement" from $\mathcal{M}$. Alternate:

1. Decode sample (w.o. replacement) from $\mathcal{M}$
2. Encode sampled element using P_X

until $\mathcal{M}$ is depleted.



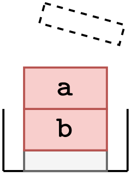
$$L(\mathcal{M}) = \varepsilon - \log \frac{1}{2/3} + \log \frac{1}{P_X(\mathtt{b})} - \log \frac{1}{1/2} + \log \frac{1}{P_X(\mathtt{a})}$$

Method: compressing $\mathcal{M}$ to $-\log P_{\mathcal{M}}(\mathcal{M})$ bits

Construct order information $H(X^n \mid \mathcal{M})$ iteratively by "sampling without replacement" from $\mathcal{M}$. Alternate:

1. Decode sample (w.o. replacement) from $\mathcal{M}$
2. Encode sampled element using P_X

until $\mathcal{M}$ is depleted.



$\{\texttt{b}\}$

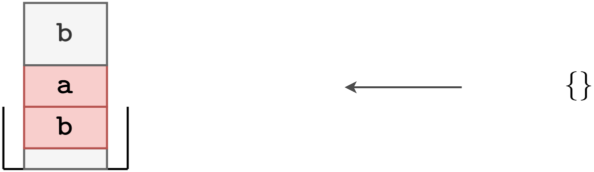
$$L(\mathcal{M}) = \varepsilon - \log \frac{1}{2/3} + \log \frac{1}{P_X(\texttt{b})} - \log \frac{1}{1/2} + \log \frac{1}{P_X(\texttt{a})} - \log \frac{1}{1/1}$$

Method: compressing $\mathcal{M}$ to $-\log P_{\mathcal{M}}(\mathcal{M})$ bits

Construct order information $H(X^n \mid \mathcal{M})$ iteratively by "sampling without replacement" from $\mathcal{M}$. Alternate:

1. Decode sample (w.o. replacement) from $\mathcal{M}$
2. Encode sampled element using P_X

until $\mathcal{M}$ is depleted.



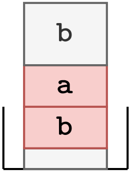
$$L(\mathcal{M}) = \varepsilon - \log \frac{1}{2/3} + \log \frac{1}{P_X(\mathtt{b})^2} - \log \frac{1}{1/2} + \log \frac{1}{P_X(\mathtt{a})} - \log \frac{1}{1/1}$$

Method: compressing $\mathcal{M}$ to $-\log P_{\mathcal{M}}(\mathcal{M})$ bits

Construct order information $H(X^n \mid \mathcal{M})$ iteratively by "sampling without replacement" from $\mathcal{M}$. Alternate:

1. Decode sample (w.o. replacement) from $\mathcal{M}$
2. Encode sampled element using P_X

until $\mathcal{M}$ is depleted.



$\{\}$

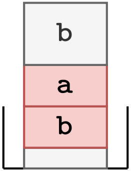
$$L(\mathcal{M}) = \varepsilon + \log \frac{1}{P_X(\mathtt{b})^2 P_X(\mathtt{a})} - \log \frac{1}{(2/3)(1/2)(1/1)}$$

Method: compressing $\mathcal{M}$ to $-\log P_{\mathcal{M}}(\mathcal{M})$ bits

Construct order information $H(X^n \mid \mathcal{M})$ iteratively by "sampling without replacement" from $\mathcal{M}$. Alternate:

1. Decode sample (w.o. replacement) from $\mathcal{M}$
2. Encode sampled element using P_X

until $\mathcal{M}$ is depleted.



$\{\}$

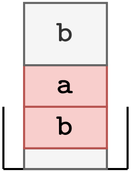
$$L(\mathcal{M}) = \varepsilon + \log \frac{1}{P_{X^n}(\mathtt{bab})} - \log \frac{1}{P_{X^n \mid \mathcal{M}}(\mathtt{bab} \mid \{\mathtt{a}, \mathtt{b}, \mathtt{b}\})}$$

Method: compressing $\mathcal{M}$ to $-\log P_{\mathcal{M}}(\mathcal{M})$ bits

Construct order information $H(X^n \mid \mathcal{M})$ iteratively by "sampling without replacement" from $\mathcal{M}$. Alternate:

1. Decode sample (w.o. replacement) from $\mathcal{M}$
2. Encode sampled element using P_X

until $\mathcal{M}$ is depleted.



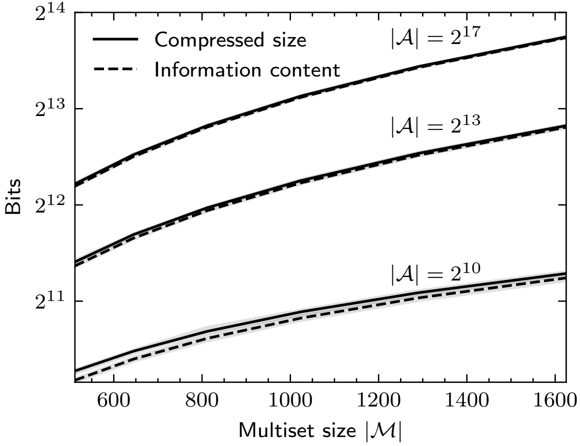
$$\{\}$$

$$L(\mathcal{M}) = \varepsilon + \log \frac{1}{P_{\mathcal{M}}(\{\mathtt{a}, \mathtt{b}, \mathtt{b}\})}$$

Experiments

Experiments: Synthetic multisets (rate)

Achieves $H(\mathcal{M}) = \mathsf{E}[-\log P_{\mathcal{M}}(\mathcal{M})]$ on single $\mathcal{M}$



Experiments: Synthetic multisets (complexity)

Average complexities

$\mathcal{O}(\log m)$ to sample from $\mathcal{M}$, where $m = \#$ unique symbols in $\mathcal{M}$

Experiments: Synthetic multisets (complexity)

Average complexities

$\mathcal{O}(\log m)$ to sample from $\mathcal{M}$, where $m = \#$ unique symbols in $\mathcal{M}$

$\mathcal{O}(p)$ to encode/decode with P_X

Experiments: Synthetic multisets (complexity)

Average complexities

$\mathcal{O}(\log m)$ to sample from $\mathcal{M}$, where $m = \#$ unique symbols in $\mathcal{M}$

$\mathcal{O}(p)$ to encode/decode with P_X

$\mathcal{O}(np + n \log m)$ total complexity to encode/decode $\mathcal{M}$

Experiments: Synthetic multisets (complexity)

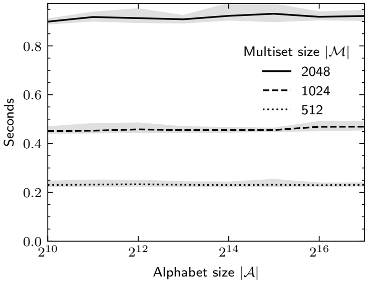
Average complexities

$\mathcal{O}(\log m)$ to sample from $\mathcal{M}$, where $m = \#$ unique symbols in $\mathcal{M}$

$\mathcal{O}(p)$ to encode/decode with P_X

$\mathcal{O}(np + n \log m)$ total complexity to encode/decode $\mathcal{M}$

Encode + decode time for fixed $m = 512$



Experiments: Synthetic multisets (complexity)

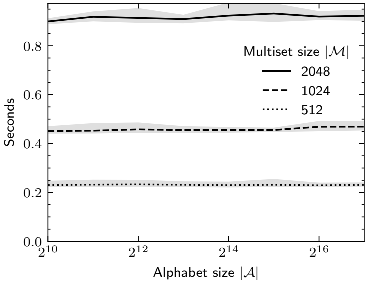
Average complexities

$\mathcal{O}(\log m)$ to sample from $\mathcal{M}$, where $m = \#$ unique symbols in $\mathcal{M}$

$\mathcal{O}(p)$ to encode/decode with P_X

$\mathcal{O}(np + n \log m)$ total complexity to encode/decode $\mathcal{M}$

Encode + decode time for fixed $m = 512$



Compute time doesn't scale with $|\mathcal{A}|$, if m is fixed

Experiments: MNIST images with WebP

Symbols X_i can be images, text, or anything else.

Experiments: MNIST images with WebP

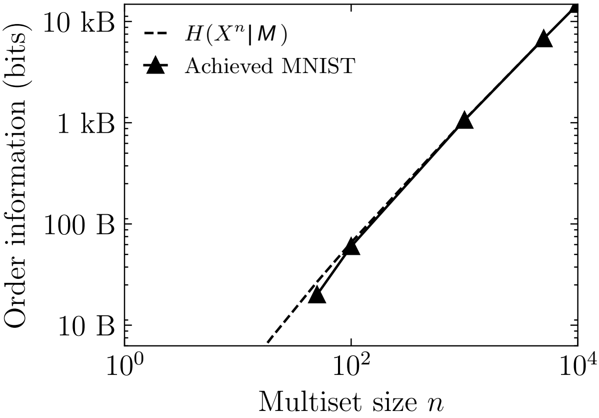
Symbols X_i can be images, text, or anything else.

Lossy codecs like WebP/JPEG can replace encoding with P_X

Experiments: MNIST images with WebP

Symbols X_i can be images, text, or anything else.

Lossy codecs like WebP/JPEG can replace encoding with P_X



Method removes all order information $H(X^n \,|\, \mathcal{M})$

Experiments: JSON maps as nested multisets

Symbols X_i can be multisets themselves (as in JSON maps)

Experiments: JSON maps as nested multisets

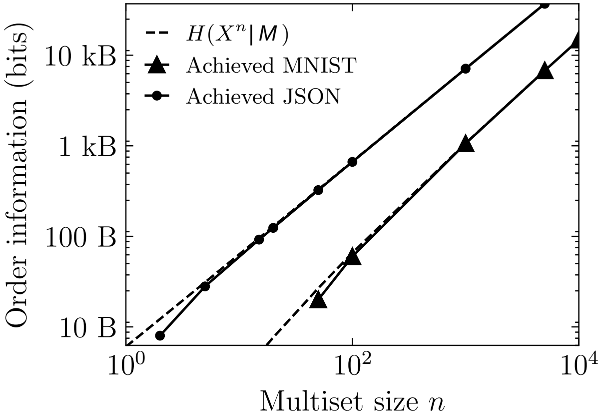
Symbols X_i can be multisets themselves (as in JSON maps)
This means $\mathcal{M}$ is a multiset of multisets

Experiments: JSON maps as nested multisets

Symbols X_i can be multisets themselves (as in JSON maps)
This means $\mathcal{M}$ is a multiset of multisets
Method naturally extends to this case

Experiments: JSON maps as nested multisets

Symbols X_i can be multisets themselves (as in JSON maps)
This means $\mathcal{M}$ is a multiset of multisets
Method naturally extends to this case



Method removes all order information $H(X^n \mid \mathcal{M})$

Conclusion

Conclusion

- Problem: encode $\mathcal{M} = \{X_1, \ldots, X_n\}$ at $H(\mathcal{M})$ losslessly

Conclusion

- Problem: encode $\mathcal{M} = \{X_1, \ldots, X_n\}$ at $H(\mathcal{M})$ losslessly
- Current methods require at least $\mathcal{O}(|\mathcal{A}|)$ compute

Conclusion

- Problem: encode $\mathcal{M} = \{X_1, \ldots, X_n\}$ at $H(\mathcal{M})$ losslessly
- Current methods require at least $\mathcal{O}(|\mathcal{A}|)$ compute
- Our method requires $\mathcal{O}(np + n \log m)$, independent of $|\mathcal{A}|$

Conclusion

- Problem: encode $\mathcal{M} = \{X_1, \ldots, X_n\}$ at $H(\mathcal{M})$ losslessly
- Current methods require at least $\mathcal{O}(|\mathcal{A}|)$ compute
- Our method requires $\mathcal{O}(np + n \log m)$, independent of $|\mathcal{A}|$
- It relies on BB-ANS: $H(\mathcal{M}) = H(X^n) - H(X^n \,|\, \mathcal{M})$

Conclusion

- Problem: encode $\mathcal{M} = \{X_1, \ldots, X_n\}$ at $H(\mathcal{M})$ losslessly
- Current methods require at least $\mathcal{O}(|\mathcal{A}|)$ compute
- Our method requires $\mathcal{O}(np + n \log m)$, independent of $|\mathcal{A}|$
- It relies on BB-ANS: $H(\mathcal{M}) = H(X^n) - H(X^n \,|\, \mathcal{M})$
- Can compress single $\mathcal{M}$ to $-\log P_{\mathcal{M}}(\mathcal{M})$ bits

Conclusion

- Problem: encode $\mathcal{M} = \{X_1, \ldots, X_n\}$ at $H(\mathcal{M})$ losslessly
- Current methods require at least $\mathcal{O}(|\mathcal{A}|)$ compute
- Our method requires $\mathcal{O}(np + n \log m)$, independent of $|\mathcal{A}|$
- It relies on BB-ANS: $H(\mathcal{M}) = H(X^n) - H(X^n \,|\, \mathcal{M})$
- Can compress single $\mathcal{M}$ to $-\log P_{\mathcal{M}}(\mathcal{M})$ bits
- Symbols can be anything (e.g. images, text, multisets)

Thank you!

Meta


Daniel Severo


James Townsend

UCL

UNIVERSITY OF TORONTO


Ashish Khisti


Alireza Makhzani


Karen Ullrich

VECTOR INSTITUTE

Presented by: dsevero.com and j-towns.github.io

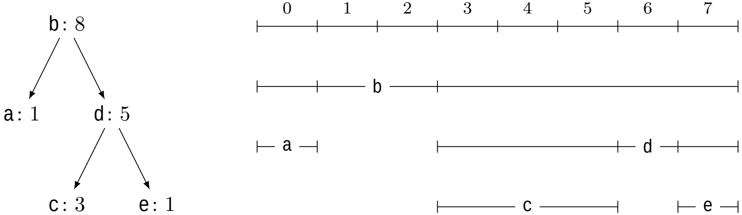
Code: github.com/facebookresearch/multiset-compression

Bonus: dynamic multiset data structure

We use something like a *Fenwick tree* for the multiset.

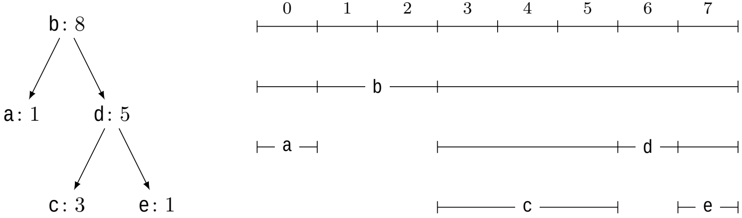
Bonus: dynamic multiset data structure

We use something like a *Fenwick tree* for the multiset.
E.g. for $\mathcal{M} = \{\mathtt{a}, \mathtt{b}, \mathtt{b}, \mathtt{c}, \mathtt{c}, \mathtt{c}, \mathtt{d}, \mathtt{e}\}$...



Bonus: dynamic multiset data structure

We use something like a *Fenwick tree* for the multiset.
E.g. for $\mathcal{M} = \{\mathtt{a}, \mathtt{b}, \mathtt{b}, \mathtt{c}, \mathtt{c}, \mathtt{c}, \mathtt{d}, \mathtt{e}\}$...



Has $O(\log n)$ insertion, deletion and F_X, P_X lookup :-).