Practical notes on lossy compression of scientific data

shooting own foot

Rostislav Kouznetsov
rostislav.kouznetsov@fmi.fi
Finnish Meteorological Institute
25.5.2022

ILMATIETEEN LAITOS
METEOROLOGISKA INSTITUTET
FINNISH METEOROLOGICAL INSTITUTE

Stories

- Precision trimming and lossy compression
- Decimal-digits ambiguous
- On information-theory methods
- Keeping absolute vs relative precision
- Acceptable error margin depends on application
- Packing real to integers – bad idea (with some exceptions)
- Bit-Shuffle not always beneficial

Information reduction + lossless compression

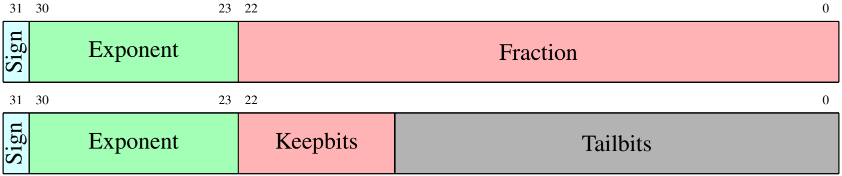
”You can get much farther with a kind word and a gun than you can with a kind word alone”

not really Al Capone

- ”Kind word” $\rightarrow$ lossless compression
- ”Gun” $\rightarrow$ precision trimming, ”packing”, or other information-reduction technique

Information reduction is **irreversible**.

Precision-trimming in IEEE-754



- Round to zero some least-significant bits
- Apply lossless compression

Error $\leqslant$ value MSB of tail bits (if proper rounding used).
Error cannot be accurately specified in terms of ”decimals”.

Maximum error can be specified as:

- Absolute (fixed value MSB of tail bits)
- Relative to a value (fixed number of keep-bits)
- Something else (e.g. as in recent nco)

Significant decimals: vague measure

```
ncks -h -4 -L5 --ppc sig53=N infile.nc outfile.nc
```

Max rel.error and compression efficiency for a ”-5/3” signal from Kouznetsov (2021)

ppc	1	2	3	4	5	6	7
nco-4.9.1 to 4.9.4: Bit grooming							
rerr	3.1e-02	3.8e-03	4.9e-04	3.0e-05	3.7e-06	3.6e-07	0.0e+00
comprr	0.10173	0.19155	0.27992	0.37821	0.51420	0.58844	0.61565
nco-4.9.5 to 5.0.2: Bit rounding							
rerr	1.5e-02	1.9e-03	2.4e-04	1.5e-05	1.9e-06	2.4e-07	0.0e+00
comprr	0.09723	0.18701	0.26817	0.37812	0.50525	0.56224	0.61565
nco-5.0.3 onwards: Granular bit rounding							
rerr	1.4e-01	1.9e-02	2.2e-03	1.4e-04	1.8e-05	2.2e-06	0.0e+00
comprr	0.07080	0.09684	0.18706	0.29790	0.37853	0.50550	0.61565

- All versions keep same number of decimals
- Since v5.0.3 $\mathrm{N}+1$ should be used instead of N in ppc
- Error margin or 'number of keep-bits' is more robust measure

Significant decimals: vague measure

```
ncks -h -4 -L5 --ppc sig53=N infile.nc outfile.nc
```

Max rel.error and compression efficiency for a ”-5/3” signal from Kouznetsov (2021)

ppc	1	2	3	**4**	5	6	7
nco-4.9.1 to 4.9.4: Bit grooming							
rerr	3.1e-02	3.8e-03	4.9e-04	**3.0e-05**	3.7e-06	3.6e-07	0.0e+00
comprr	0.10173	0.19155	0.27992	**0.37821**	0.51420	0.58844	0.61565
nco-4.9.5 to 5.0.2: Bit rounding							
rerr	1.5e-02	1.9e-03	2.4e-04	**1.5e-05**	1.9e-06	2.4e-07	0.0e+00
comprr	0.09723	0.18701	0.26817	**0.37812**	0.50525	0.56224	0.61565
nco-5.0.3 onwards: Granular bit rounding							
rerr	1.4e-01	1.9e-02	2.2e-03	**1.4e-04**	1.8e-05	2.2e-06	0.0e+00
comprr	0.07080	0.09684	0.18706	**0.29790**	0.37853	0.50550	0.61565

- All versions keep same number of decimals
- Since v5.0.3 $\mathrm{N}+1$ should be used instead of N in ppc
- Error margin or 'number of keep-bits' is more robust measure

Significant decimals: vague measure

```
ncks -h -4 -L5 --ppc sig53=N infile.nc outfile.nc
```

Max rel.error and compression efficiency for a "-5/3" signal from Kouznetsov (2021)

ppc	1	2	3	**4**	5	6	7
nco-4.9.1 to 4.9.4: Bit grooming							
rerr	3.1e-02	3.8e-03	4.9e-04	**3.0e-05**	3.7e-06	3.6e-07	0.0e+00
comprr	0.10173	0.19155	0.27992	**0.37821**	0.51420	0.58844	0.61565
nco-4.9.5 to 5.0.2: Bit rounding							
rerr	1.5e-02	1.9e-03	2.4e-04	**1.5e-05**	1.9e-06	2.4e-07	0.0e+00
comprr	0.09723	0.18701	0.26817	**0.37812**	0.50525	0.56224	0.61565
nco-5.0.3 onwards: Granular bit rounding							
rerr	1.4e-01	1.9e-02	2.2e-03	**1.4e-04**	1.8e-05	2.2e-06	0.0e+00
comprr	0.07080	0.09684	0.18706	**0.29790**	0.37853	0.50550	0.61565

- All versions keep same number of decimals
- Since v5.0.3 $\mathrm{N}+1$ should be used instead of N in ppc
- Error margin or 'number of keep-bits' is more robust measure

Significant decimals: vague measure

```
ncks -h -4 -L5 --ppc sig53=N infile.nc outfile.nc
```

Max rel.error and compression efficiency for a ”-5/3” signal from Kouznetsov (2021)

ppc	1	2	3	**4**	5	6	7
nco-4.9.1 to 4.9.4: Bit grooming							
rerr	3.1e-02	3.8e-03	4.9e-04	**3.0e-05**	3.7e-06	3.6e-07	0.0e+00
comprr	0.10173	0.19155	0.27992	**0.37821**	0.51420	0.58844	0.61565
nco-4.9.5 to 5.0.2: Bit rounding							
rerr	1.5e-02	1.9e-03	2.4e-04	**1.5e-05**	1.9e-06	2.4e-07	0.0e+00
comprr	0.09723	0.18701	0.26817	**0.37812**	0.50525	0.56224	0.61565
nco-5.0.3 onwards: Granular bit rounding							
rerr	1.4e-01	1.9e-02	2.2e-03	1.4e-04	**1.8e-05**	2.2e-06	0.0e+00
comprr	0.07080	0.09684	0.18706	0.29790	**0.37853**	0.50550	0.61565

- All versions keep same number of decimals
- Since v5.0.3 $\mathrm{N}+1$ should be used instead of N in ppc
- Error margin or ’number of keep-bits’ is more robust measure

Bit-information (Klöwer etal 2021, Nat.Comp.Sci.)

Problem

We do not know the error margin, but want to apply a lossy compression

Method

1. *Evaluate mutual information in adjacent cells for each bit*
2. *Trim precision to keep 99% of information*

Bit-information (Klöwer etal 2021, Nat.Comp.Sci.)

Problem

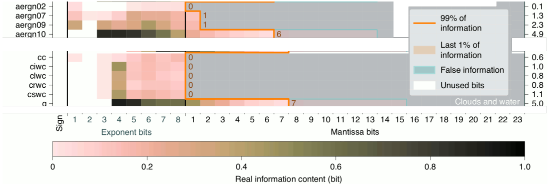
We do not know the error margin, but want to apply a lossy compression

Method

1. *Evaluate mutual information in adjacent cells for each bit*
2. *Trim precision to keep 99% of information*

Example: Cloud cover

Bit information in CAMS variables:



Adapted from Klöwer etal (2021, Fig. 2)

- 0 mantissa bits needed (Fig. 2 says so)
- Can be 1, 0.5, 0.25, 0.125, etc.
- 0.001 and 0.002 are different
- **1 and 0.8 are the same**
- Is it really the precision of cc in IFS?

Bit-information (Klöwer etal 2021, Nat.Comp.Sci.)

Problem

We do not know the error margin, but want to apply a lossy compression

Method

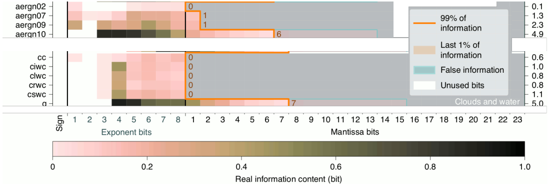
1. *Evaluate mutual information in adjacent cells for each bit*
2. *Trim precision to keep 99% of information*

Assumptions (missing from the paper)

1. *Relative* error model applicable
2. Statistical properties are homogeneous
3. Needed info is not within the lost 1%

Example: Cloud cover

Bit information in CAMS variables:



Adapted from Klöwer etal (2021, Fig. 2)

- 0 mantissa bits needed (Fig. 2 says so)
- Can be 1, 0.5, 0.25, 0.125, etc.
- 0.001 and 0.002 are different
- **1 and 0.8 are the same**
- Is it really the precision of cc in IFS?

Conclusions

- Lossy compression introduces distortion
- Trading precision for size reduction
- ”Decimal digits” is ambiguous measure of precision:
  - Data representation is **binary**
  - Decimal digits do not map to binary
  - Explicit error margin (or keepbits) is better
- Knowing the data and acceptable error margins is essential
  - Ensuring that a smart method is applicable needs even more knowledge
  - No silver bullet
- Powerful tool implies awareness and responsibility

More info: R. Kouznetsov (2021) Geosci Mod Dev
`https://doi.org/10.5194/gmd-14-377-2021`

Absolute vs relative precision: trim both!

Common model for error: $|\Delta x| \leqslant \max(A|x|, B)$

- A – relative error (fraction)
- B – absolute error (units of x)

Relative precision:

- Select the right number of keepbits:

$$\text{keepbits} = \lceil \log_2(1/A) \rceil - 1$$

- Apply bit rounding

Absolute precision:

- Select *binary* quantum as $\tilde{B} = 2^M$: $M = \lfloor \log_2(B) \rfloor + 1$
- Multiply data with $\tilde{B}$
- Round to nearest integer
- Divide data with $\tilde{B}$ (as float)

- Trimmings can be applied in any order
- Nicely quantified error margin
- See Kouznetsov (2021, Geosci Model Devel) for implementation

Packing floats to to integers: bad practice

Implementation for NetCDF:

- Map the data range to the range of integers
- Usually 16-bit integer (-32768. . . 32767)
- Quantum: 1/65535 of the field range
- `add_offset` and `scale_factor` attributes!

	Packing	Absolute precision
Error margin	Decided by data	Explicitly specified
Monotonous	NO	YES
Reading to float	Needs unpacking	Transparent
Zero value maps to	Something small	Zero
Keeps sign	NO	YES
Concatenating arrays	Needs repacking	YES
Keeps integral	Sometimes	For not-too-exotic fields
Compressed size	Similar at the same error margin	

Only excuse to apply packing – compatibility with old standards, e.g. GRIB

Eccodes allow for explicit absolute precision specification.

On acceptable error in model data

- If proper rounding and enough variability avraging cancels out
- Normal error propagation rules apply

Model-measurement comparison

- Evaluation, regulatory application etc...
- Philosophical concept:
  ”unobservable” = ”non-existent”
- Error should be less than one for the best imaginable observation
- Example: Air Quality forecast:
  - Scores (RMSE, corr., etc.) same for 4 and 5 keep-bits
  - Median timeseries with 5 bits look bad
  - We keep 7 bits

Gradients, divergencies, deltas etc.

- Absolute error margin
- Acceptable error for resulting delta
- Depends on the resolution

Other applications

- Depends on purpose of modelling

On shuffling

Shuffling: alter the storage order of data to improve compression efficiency.

- In multi-byte data types most-significant byte changes slowly
- In trimmed-precision data least-significant byte(s) have much in common
- Idea: To group “similar” bytes, to improve compression
- Can gain $\sim$ x2 better compression
- Applied by nco to *all* compressed variables since 3_9_0 (2007)

- Sparse fields used for
  - Gridded station observations
  - Emissions of point sources
  - Wild-land fires
  - etc.
- No common way to store sparse data

- Regular 2D fields + compression
- Compression factors ($\sim$x100) makes feasible
- Without shuffling yet another x2 can be achieved

NetCDF vs GRIB(ccsds)

SILAM (`http://silam.fmi.fi`) use-case:

- Offline chemistry-transport model
- Reads during a run: meteorology, chemical boundaries, emission fields
- A process needs sub-domain
- Many threads available for decompression
- Many fields in different grids
- I/O limits scalability

	NetCDF4	GRIB2 (ccsds)
Data structure	Any array	2D fields that have codes
Random access	YES	Via external index
Floating-point	Yes	Packed to unsigned int, arbitrary bits
Grid sub-setting	If chunked properly	NO
Multi-thread decompression	NO	YES
File-system latency	Important	Less important
Multi-process I/O at HPFS	Bottleneck	Single reader