Predictive coding

Another important method used to achieve compression is the one of the predictive approach. It is based on the fact that the data to compress **does not differ much from one value to the other**, so they **can be predicted** well enough from the previous values. Therefore, predictive methods are applied especially in the case where the data are in fact time series (the one-dimensional case, i.e. audio signals) or space series (the two-dimensional case, i.e. images).

Even if in the following we will present those methods in the context of compression, we will use the term predictive coding, because in the scientific literature it is mostly used so.

1. General predictive coding-decoding scheme

The compression algorithm based on predictive coding works as follows (Figure 1):

1. for each input value **a value is predicted** based on a prediction scheme;
2. the **prediction error** is computed as the **difference** between the original value and the predicted value;
3. the prediction error is sent to an **entropic coder** (Huffman , arithmetic etc.) and after that saved in the compressed stream.

Decompression works symmetrical to the compression (Figure 2):

1. a value is taken from the compressed stream and is **entropic decoded** so that the **prediction error** is obtained;
2. based on the **same prediction scheme** the **current value is predicted**;
3. the prediction error **is added** to the predicted value so that the original value is obtained and saved in the decompressed stream.

In order to the scheme to work correctly it is essential that the prediction is made only based on values previously coded, because only these are available to the decoder (prediction has to be done only based on the „past”). This condition is necessary for the two predictors to work identical (to produce the same predicted value).

As described previously (the prediction is substracted at the coder and added at the decoder) the method falls in the lossless class.

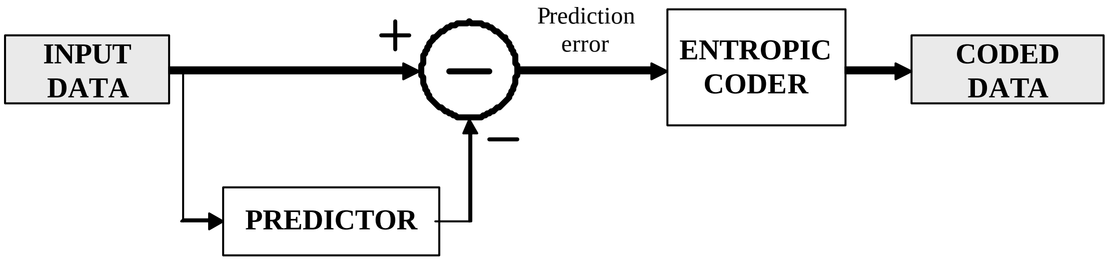


Figure 1 General coding scheme using predictive techniques

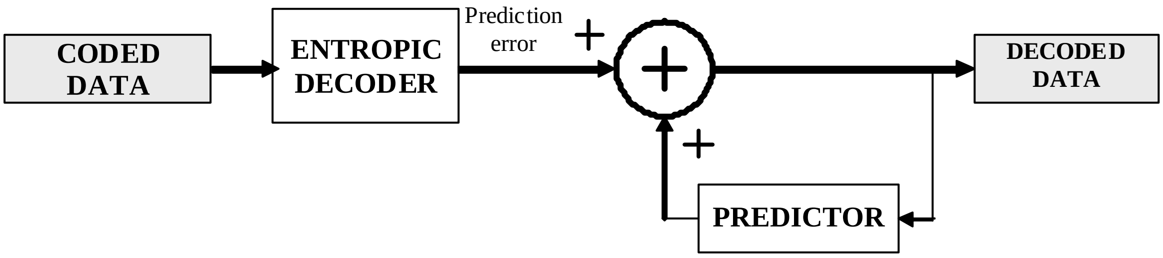


Figure 2 General decoding scheme using predictive techniques

2. Predictive coding in the one-dimensional case (time series) - Example

Let's consider the series:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Initial value VI	0	4	7	9	10	9	7	4	0	-4	-7	-9	-10	-9	-7	-4	0	4
Predicted value VP	**0**	0	4	7	9	10	9	7	4	0	-4	-7	-9	-10	-9	-7	-4	0
Prediction error EP	0	4	3	2	1	-1	-2	-3	-4	-4	-3	-2	-1	1	2	3	4	4

As we can notice, the series represents the sampling of a sinusoidal signal. By index we have labeled the (time) index in the series. In the example the A predictor was considered (it predicts with the previously value from the series).

...	B	A	?	...

Step by step we compute for each index k:

```
VP[k] = VI[k-1]          // prediction
EP[k] = VI[k] - VP[k]    // prediction error
```

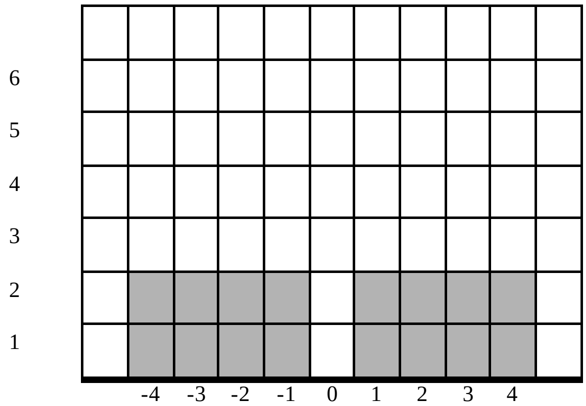
At the beginning, when we can not apply the chosen predictor (the previous value does not exist) we have to make some assumptions regarding prediction, i.e. `VP[0] = 0` is considered.

The previous example is repeated for the `2A-B` predictor. It can be interpreted as `A+(A-B)`, meaning the extension of the segment that goes through A and B. After similar computing we get:

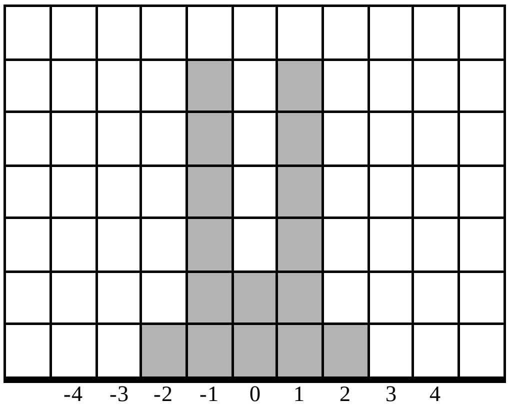
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Initial value VI	0	4	7	9	10	9	7	4	0	-4	-7	-9	-10	-9	-7	-4	0	4	7
Predicted value VP	**0**	**0**	8	10	11	11	8	5	1	-4	-8	-10	-11	-11	-8	-5	-1	4	8
Prediction error EP	0	4	-1	-1	-1	-2	-1	-1	-1	0	1	1	1	2	1	1	1	0	-1

In that case for the first two predicted values we consider `VP[0]=0, VP[1]=VI[0]=0`.

In order to compare the performances of the 2 predictors we represent the prediction error histogram.



Predictor A



Predictor 2A-B

We have taken into consideration in each case only the first 16 values of the prediction error corresponding to the predictions effectively made by the `A` and `2A-B` respectively (not the special cases from the beginning).

We notice that for the `2A-B` predictor the histogram is more unbalanced toward the 0 value. That will allow the entropic coder to get better compression results. But both histograms are unbalanced toward the 0 value compared to the histogram of the original data.

At the decoding we start only from the string of the prediction error. We compute then **step by step** for each index:

```
VP[k] = VD[k-1]          // prediction
VD[k] = VP[k] + VP[k]    // reconstruction
```

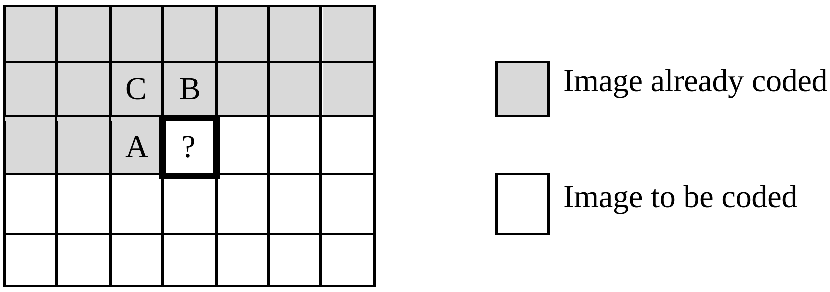
Obviously, at the beginning, when we can not apply the chosen predictor, we make the same assumptions as the coder. We present the decoded values corresponding to the `2A-B` predictor.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Prediction error EP	0	4	-1	-1	-1	-2	-1	-1	-1	0	1	1	1	2	1	1	1	0	-1
Predicted value VP	**0**	**0**	8	10	11	11	8	5	1	-4	-8	-10	-11	-11	-8	-5	-1	4	8
Decoded value VD	0	4	7	9	10	9	7	4	0	-4	-7	-9	-10	-9	-7	-4	0	4	7

We notice that data was restored without errors, as expected.

3. Predictive coding in the two-dimensional case (images)

In the one-dimensional case the method relies on the correlation of the neighboring values and, therefore, on the predictability of the next value along the time axis. In the two-dimensional case correlations exist along both axes, so that image predictors must take this into account.



In this case prediction is realized (almost always) pixel by pixel from left to right and from top to bottom. Prediction can be made by any rule, but only based on the pixels in the gray area (the image already processed). This restriction is necessary for the predictors to work identically (to generate the same prediction value).

We present in the following the most important prediction rules used in this context.

Prediction rules used in JPEG standard

JPEG standard uses for the lossless mode (completely different to the lossy mode, based on DCT) a standard predictive approach and the following 7 predictors:

```
1:  A
2:  B
3:  C
4:  A + B - C
5:  A + (B - C) / 2
6:  B + (A - C) / 2
7:  (A + B) / 2
```

The first 3 are one-dimensional and the last 4 are really two-dimensional. The A, B and C notations are the one described in the previous figure.

Prediction rule from LOCO-I (standardized in JPEG-LS)

In this case the predicted value is given by:

```
min(A,B)      if C >= max(A,B)
max(A,B)      if C <= min(A,B)
A+B-C         otherwise
```

The predictor is designed to select B when a vertical edge exists in that point, A when a horizontal edge exists and `A+B-C` if no edge is detected. The last type of prediction is similar to considering that the point is in the plane defined by the three pixels A B and C.

This predictor was used in data prediction under different interpretations. The most important one is that of mediation of the three predictors `A`, `B` and `A+B-C`. Combining those interpretations the predictor was named in the standardization process “Median Edge Detector”.

Prediction rule of CALIC

The prediction rule of the CALIC method is presented in the following:

```
IF (d_v+d_h>32)
       I*[i,j] = (d_v*I[i-1,j]+d_h*I[i,j-1])/(d_v+d_h)+(I[i+1,j-1]-I[i-1,j-1])/8
ELSE IF (d_v-d_h>12)
       I*[i,j] = (2*I[i-1,j]+I[i,j-1])/3+(I[i+1,j-1]-I[i-1,j-1])/8
ELSE IF (d_h+d_v>12)
       I*[i,j] = (I[i-1,j]+2*I[i,j-1])/3+(I[i+1,j-1]-I[i-1,j-1])/8
ELSE
       I*[i,j] = (I[i-1,j]+I[i,j-1])/2+(I[i+1,j-1]-I[i-1,j-1])/8

IF (d_45-d_135>32)
       I*[i,j] = I*[i,j]+(I[i+1,j-1]-I[i-1,j-1])/8
ELSE IF (d_45-d_135>16)
       I*[i,j] = I*[i,j]+(I[i+1,j-1]-I[i-1,j-1])/16
ELSE IF (d_135-d_45>32)
       I*[i,j] = I*[i,j]+(I[i-1,j-1]-I[i+1,j-1])/8
ELSE IF(d_135-d_45>16)
       I*[i,j] = I*[i,j]+(I[i-1,j-1]-I[i+1,j-1])/16
```

where `I[i,j]` represents the position that is analyzed at the current moment, `I*[i,j]` represents the value of the prediction and d_{45}, d_{135}, d_v si d_h represents:

```
d_h    =|I[i-1,j  ]-I[i-2,j  ]|+|I[i  ,j-1]-I[i-1,j-1]|+|I[i+1,j-1]-I[i  ,j-1]|
d_v    =|I[i-1,j  ]-I[i-1,j-1]|+|I[i  ,j-1]-I[i  ,j-2]|+|I[i+1,j-1]-I[i+1,j-2]|
d_45   =|I[i-1,j  ]-I[i-2,j-1]|+|I[i-1,j-1]-I[i-2,j-2]|+|I[i  ,j-1]-I[i-1,j-2]|
d_135  =|I[i+1,j-1]-I[i+2,j-2]|+|I[i  ,j-1]-I[i+1,j-2]|+|I[i  ,j-1]-I[i-1,j  ]|
```

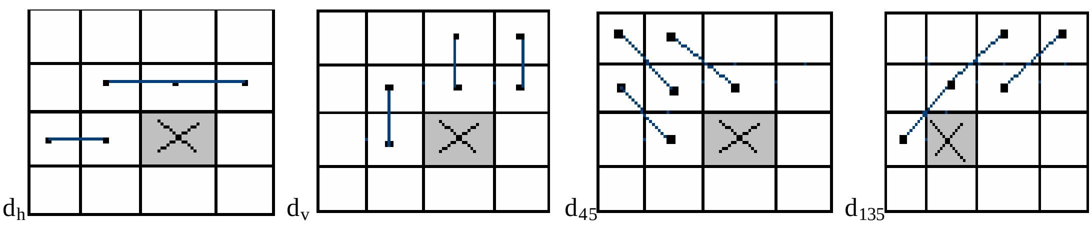


Figure 3 Pixels used in computing of the values d_h, d_v, d_{45}, d_{135}

As we can notice, the horizontal, vertical and 45º and respectively 135º gradient is computed (as presented in Figure 3) and, based on these, a correction is made to the previously predicted value.

Remark to predictive coding of images

In the cases where the two-dimensional prediction rule can not be applied because of the lack of the needed pixels a different approach has to be established (with the decoder). This happens usually on the first pixel (where a fixed prediction is used), on the first line (where the A predictor can be chosen) and on the first column (where the B predictor can be chosen).

4. Remarks

Regarding the predictive coding we have to make some important remarks:

1. By applying prediction the **dynamic range of the signal grows** as follows:

Input signal range		Output signal range
[0 ÷ N]	=>	[-N ÷ +N]
[-N ÷ +N]	=>	[-2N ÷ +2N]

needing for representation **one more bit**. Therefore, by prediction, there is no compression at all, but even a growth of representation dimension. Yet, this is compensated by the unbalancing of the histogram toward 0, so that the entropy coder that follows can obtain a much better compression.

2. In predictive coding the problem is **not one of binary classification HIT / MISS** regarding the prediction value (as usual in hardware prediction). A "better" prediction differs from a "worse" one by the way it unbalances the histogram toward 0 and allows the entropic coder to realize a higher compression ratio.

3. Entropy coding that follows the prediction step is realized usually in a contextual way, having **one statistic model for each possible context**, context being defined also based on the pixels used in prediction (the context problem is not detailed here).

4. The predictive method does not impose a specific prediction rule, so that **any kind of prediction rules**, even very **complicated** (high degree polynomials, neural predictors, etc.) can be used, but adapted to the data specific.