Datenkompression:

Differentialcodierung

H. Fernau

email: fernau@uni-trier.de

SoSe 2013

Universität Trier

Differentialcodierung—Allgemeines

Grundidee der *Differentialcodierung*:
nutze Korrelation aufeinander folgender Quellenausgaben dadurch aus, dass anstelle der Originalwerte die Differenzen zweier benachbarter Werte (codiert) übertragen werden.

Hoffnung: Verkleinerung von Wertebereich und Varianz.

Ein erstes Beispiel: eine Abtastung der Kosinusfunktion

$$f(t) = \cos(\frac{\pi}{6}t), \quad t \in [0, 6].$$

Die Werte $x_n = f(n/10)$, $n = 0, \ldots 60$ liegen in $[-1, +1]$.
Hingegen liegen die Differenzen $d_n = x_n - x_{n-1}$ in $[-0,2; 0,2]$.*
Benutzt man einen 2-Bit-Gleichquantisierer, so führt dies bei Übertragung der x_n zu einer Schrittweite von $2 \times 1/2^2 = 0,5$, und der betreffende Quantisierfehler liegt im Intervall $[-0,25; 0,25]$; bei Übertragung der d_n nimmt man die Schrittweite $0,1$
$\rightsquigarrow$ Quantisierfehler beschränkt auf $[-0,05; 0,05]$.

Bei Bildern führt der Übergang zu Pixeldifferenzen oft zu Histogrammen, die der Laplace-Verteilung ähneln. Dies berücksichtigend ist eine Quantisierung des „Differenzenbildes" mit fünf oder gar vier Bit genauso gut wie die des Originalbilds mit sieben Bit.

*Eine Ausnahme bildet der erste Wert $d_0 = x_0$.

Das **Phänomen der** *Fehlerfortpflanzung*
bei verlustbehafteter Übertragung wurde bisher vernachlässigt, was an folgendem Beispiel deutlich wird:

Folge $\{x_n\}$	6,2	9,7	13,2	5,9	8,0	7,4	4,2	1,8
Diff. $\{d_n\}$	6,2	3,5	3,5	-7,3	2,1	0,6	-3,2	-2,4
Quant. $\{\widehat{d}_n\}$	6	4	4	-6	2	0	-4	-2
Rekonstr. $\{\widehat{x}_n\}$	6	10	14	8	10	10	6	4
Abweichung	0,2	-0,3	-0,8	-2,1	-2	-2,6	-1,8	-2,2

Hierbei benutzen wir einen Quantisierer mit sieben Ausgabewerten

$$\{-6, -4, -2, 0, 2, 4, 6\}.$$

Wie man sieht, scheint die Abweichung tendenziell immer größer zu werden.

Fehlerfortpflanzung formaler untersucht:

Bezeichnungen: $\{x_n\}$: Ausgangsfolge, $\{d_n\}$: Differenzenfolge, $\{\hat{d}_n\}$: quantisierte Differenzenfolge, $\{\hat{x}_n\}$: rekonstruierte Folge.
Startwert sei x_0, und es gelte $\hat{x}_0 = x_0$. Dann gilt:

Lemma 1 $\hat{x}_n = x_n + \sum_{k=1}^{n} q_k$.

Beweis:

$$\begin{aligned} d_n &= x_n - x_{n-1}, \text{ falls } n > 0 \\ \hat{d}_n &= Q[d_n] = d_n + q_n, \ q_n \text{ ist der Quantisierfehler} \\ \hat{x}_n &= \hat{x}_{n-1} + \hat{d}_n, \text{ falls } n > 0 \\ &= x_n + \sum_{k=1}^{n} q_k, \end{aligned}$$

denn $\hat{x}_n - \hat{x}_{n-1} = \hat{d}_n = x_n - x_{n-1} + q_n$ liefert mit $\hat{x}_0 = x_0$ per Teleskopsumme:

$$(\hat{x}_n - \hat{x}_{n-1}) + (\hat{x}_{n-1} - \hat{x}_{n-2}) + \ldots + (\hat{x}_1 - x_0) =$$

$$(x_n - x_{n-1}) + (x_{n-1} - x_{n-2}) + \ldots + (x_1 - x_0) + \sum_{k=1}^{n} q_k \, .$$

Verhinderung der Fehlerfortpflanzung

Eine leichte Modifikation verhindert diese sich im Term $\sum_{k=1}^{n} q_k$ ausdrückende Fehlerfortpflanzung:
setze nun $d_n = x_n - \widehat{x}_{n-1}$; damit gilt:

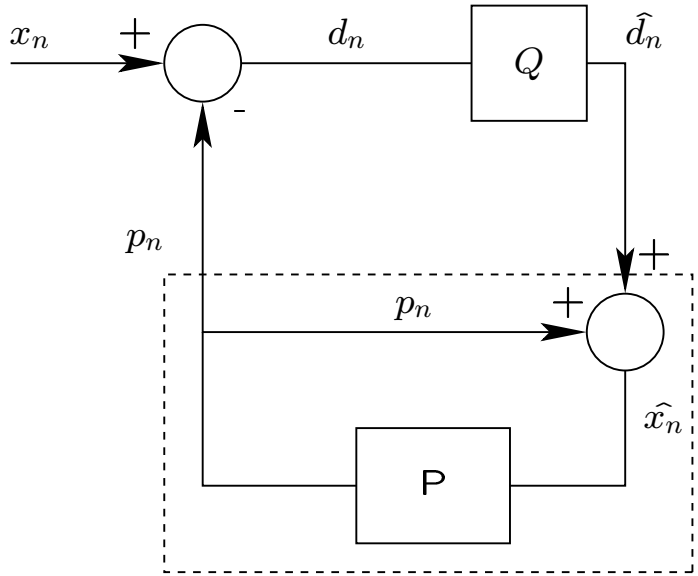
$$\widehat{x}_n = \widehat{x}_{n-1} + \widehat{d}_n = \underbrace{\widehat{x}_{n-1} + d_n}_{x_n} + q_n \,.$$

Hierbei kann $\widehat{x}_{n-1}$ als ein Schätzer für x_n interpretiert werden.
Diese Idee wird im folgenden verallgemeinert.

Unser Beispiel ohne Fehlerfortpflanzung

Folge $\{x_n\}$	6,2	9,7	13,2	5,9	8,0	7,4	4,2	1,8
Diff. $\{d_n\}$	6,2	3,7	3,2	-8,1	0	-0,6	-3,8	-2,2
Quant. $\{\widehat{d}_n\}$	6	4	4	-6	0	0	-4	-2
Rekonstr. $\{\widehat{x}_n\}$	6	10	14	8	8	8	4	2
Abweichung	0,2	-0,3	-0,8	-2,1	0	-0,6	0,2	-0,2

Prädikative Differentialcodierung (DPCM)



(Bell Labs, um 1950)

Schema eines Codierers im *prädikativen Differentialverfahren*, englisch „differential pulse code modulation" genannt, kurz *DPCM*.

Q: der zur Codierung der Differenzenfolge verwendeten Quantisierer

P: der *Prädikator* oder *Vorhersager* genannte Schätzer für x_n.

Während zuvor $p_n = \hat{x}_{n-1}$ galt, setzt man nun $p_n = f(\hat{x}_{n-1}, \hat{x}_{n-2}, \ldots, \hat{x}_0)$ an.

Bemerkenswert ist, dass der Codierer den Decodierer (im gestrichelt eingekästelten Bereich) simuliert.

Wie ist die Vorhersagefunktion f zu wählen?

Da für die Differenzenfolge $d_n = x_n - p_n$ gilt, beeinflusst die Schätzfolge p_n offenbar u. a. die Varianz σ_d^2 der Differenzfolge, die möglichst klein sein sollte.

Satz 2 *Unter den Annahmen*

1. $p_n = f(\hat{x}_{n-1}, \hat{x}_{n-2}, \ldots, \hat{x}_0) \sim f(x_{n-1}, x_{n-2}, \ldots, x_0)$ *und*

2. $\{x_n\}$ *ist stationär*

kann man zeigen, dass σ_d^2 *durch den bedingten Erwartungswert*

$$E[x_n \mid x_{n-1}, \ldots, x_0]$$

minimiert wird.

Vereinfachungen für Prädikatoren:

Immer noch zu kompliziert ?! $\rightsquigarrow$
Annahme: Vorhersagefunktion ist linear, d. h. $p_n = \sum_{i=1}^{N} a_i \widehat{x}_{n-i}$;
N heißt auch *Ordnung* des Prädikators.
Erste Voraussetzung im Satz $\rightsquigarrow$

$$\sigma_d^2 = E[(x_n - \sum_{i=1}^{N} a_i x_{n-i})^2] \quad \text{ist zu minimieren.}$$

Hierzu setzen wir die partiellen Ableitungen bezüglich der a_i Null:

$$\begin{aligned}
\frac{\partial \sigma_d^2}{\partial a_1} &= -2E\left[\left(x_n - \sum_{i=1}^{N} a_i x_{n-i}\right) x_{n-1}\right] = 0 \\
\frac{\partial \sigma_d^2}{\partial a_2} &= -2E\left[\left(x_n - \sum_{i=1}^{N} a_i x_{n-i}\right) x_{n-2}\right] = 0 \\
&\vdots \qquad\qquad \vdots \\
\frac{\partial \sigma_d^2}{\partial a_N} &= -2E\left[\left(x_n - \sum_{i=1}^{N} a_i x_{n-i}\right) x_{n-N}\right] = 0.
\end{aligned}$$

Unter Benutzung der *Autokorrelationsfunktion*

$$R_{xx}(k) = E[x_n x_{n+k}]$$

können wir also ansetzen:

$$\begin{aligned}
\sum_{i=1}^{N} a_i R_{xx}(i-1) &= R_{xx}(1) \\
\sum_{i=1}^{N} a_i R_{xx}(i-2) &= R_{xx}(2) \\
&\vdots \quad\quad \vdots \\
\sum_{i=1}^{N} a_i R_{xx}(i-N) &= R_{xx}(N)
\end{aligned}$$

Dies lässt sich auch in **Matrixform** schreiben:

$$\mathbf{RA} = \mathbf{P}$$

wobei

$$\mathbf{R} = \begin{bmatrix} R_{xx}(0) & R_{xx}(1) & R_{xx}(2) & \dots & R_{xx}(N-1) \\ R_{xx}(1) & R_{xx}(0) & R_{xx}(1) & \dots & R_{xx}(N-2) \\ R_{xx}(2) & R_{xx}(1) & R_{xx}(0) & \dots & R_{xx}(N-3) \\ \vdots & \vdots & \vdots & & \vdots \\ R_{xx}(N-1) & R_{xx}(N-2) & R_{xx}(N-3) & \dots & R_{xx}(0) \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_N \end{bmatrix} \qquad \mathbf{P} = \begin{bmatrix} R_{xx}(1) \\ R_{xx}(2) \\ R_{xx}(3) \\ \vdots \\ R_{xx}(N) \end{bmatrix}$$

Mit den soeben eingeführten Notationen gilt:

Korollar 3 *Unter den Annahmen von Satz 2 und für lineare Prädikatorfunktionen ergibt sich der Vektor der Prädikatorkoeffizienten durch:*

$$\mathbf{A} = \mathbf{R}^{-1}\mathbf{P}$$

Natürlich können wir in der Praxis nicht annehmen, wir hätten bereits Kenntnis der genauen Autokorrelationswerte, in der *Autokorrelationsmatrix* $\mathbf{R}$ zusammengefasst. $\rightsquigarrow$ Verwende Schätzer dieser Werte M Datenpunkte:

$$R_{xx}(k) = \frac{1}{M-k} \sum_{i=1}^{M-k} x_i x_{i+k} \,.$$

Leistungswerte von DPCM-Systemen mit unterschiedlichen Quantisierern und Prädikatoren:

Quantisierer	Prädikatorordnung	SNR (dB)	SPER (dB)
2-Bit	keine	2,43	
	1	3,37	2,65
	2	8,35	5,9
	3	8,74	6,1
3-Bit	keine	3,65	
	1	3,87	2,74
	2	9,81	6,37
	3	10,16	6,71

DPCM in der Sprachverarbeitung

Für eine Spracheingabe wurden auf diese Weise die Prädikatorkoeffizienten ermittelt. Diese sind natürlich von der Ordnung des Prädikators abhängig. Die Tabelle enthält das Verhältnis SNR von Signal zur Verzerrung sowie das *Verhältnis SPER von Signal zur Vorhersage*, d. i.,

$$SPER(dB) = \frac{\sum_{i=1}^{M} x_i^2}{\sum_{i=1}^{M} (x_i - p_i)^2}$$

für dieses Sprachbeispiel, bei dem die Quelle (zu Vergleichszwecken) einmal mit einem 2-(bzw. 3-)Bit Quantisierer codiert wurde (daher keine Prädikatorordnung) und dann mit einem linearen Prädikator (der gemäß dem Ansatz von Kor. 3 erhalten wurde) der Ordnung $N = 1, 2, 3$.

Gerade bei Sprache ist die Stationaritätsannahme sicher falsch, was es nahelegt, adaptiv zu quantisieren.

Adaptive Differentialcodierung

Grundschema der Differentialcodierung $\rightsquigarrow$ es gibt zwei Komponenten, die adaptiv sein können, nämlich der Quantisierer Q und der Prädikator P.

Die weiter erkennbare Rückkopplung lässt eine Voradaptierung des Quantisierers nicht sinnvoll erscheinen.

Die Rückadaptierung von Q kann mit einem Jayant-Quantisierer geschehen.

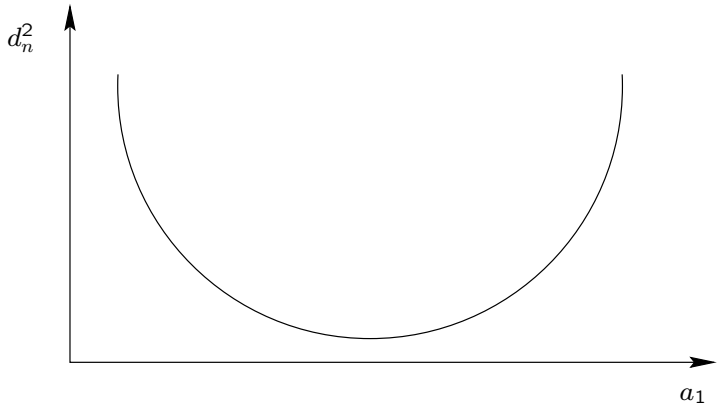
Z.B. bei Spracheingabe erscheint eine Adaptierung der Prädikatorparameter überdies sinnvoll.

Prädikatoradaptierung

Bei DPCM mit *Voradaptierung des Prädikators* (DPCM-APF) wird die Eingabe in Blöcke unterteilt, der Schätzer der Autokorrelationskoeffizienten je Block berechnet und als Begleitinformation übertragen. Insbesondere bei Sprachübertragung kann DPCM-APF nicht tolerierbare Verzögerungen bewirken.

Wir beschäftigen uns daher jetzt mit der *Rückadaptierung des Prädikators* (DPCM-APB).

Fehlerfunktion in Abhängigkeit vom Prädikatorparameter



Der Fehler eines Prädikators erster Ordnung lässt sich durch

$$d_n^2 = (x_n - a_1 \widehat{x}_{n-1})^2$$

messen. Fassen wir d_n^2 als Funktion von a_1 auf, so ergibt sich eine Parabel wie in obigem Bild.

Was ergibt sich daraus?

Ist der Wert $a_1^{(n)}$ weit vom Optimum entfernt, so ist eine raschere Adaptierung erwünscht als wenn fast das Optimum erreicht ist. Diesen Effekt erreichen wir durch Anwendung der Formel

$$\begin{aligned} a_1^{(n+1)} &= a_1^{(n)} - \alpha \frac{\partial d_n^2}{\partial a_1} \text{ mit} \\ \frac{\partial d_n^2}{\partial a_1} &= -2(x_n - a_1 \widehat{x}_{n-1})\widehat{x}_{n-1} \\ &= -2 d_n \widehat{x}_{n-1}; \text{ also ist:} \\ a_1^{(n+1)} &= a_1^{(n)} + \alpha' d_n \widehat{x}_{n-1} \end{aligned}$$

für eine die Adaptivität beeinflussende positive Konstante α bzw. α'. Da die nichtquantisierten Werte d_n nur dem Codierer zur Verfügung stehen, ersetzen wir in jener Formel d_n durch $\widehat{d}_n$:

$$a_1^{(n+1)} = a_1^{(n)} + \alpha \widehat{d}_n \widehat{x}_{n-1}$$

Codierer höherer Ordnung

Für einen Codierer höherer Ordnung $N > 1$ erhält man ganz entsprechend durch Betrachtung von

$$d_n^2 = \left(x_n - \sum_{i=1}^{N} a_i \widehat{x}_{n-i} \right)^2$$

sowie der partiellen Ableitungen dieser Funktion nach den a_j:

Lemma 4 (Adaptionsregel für den j-ten Prädikatorkoeffizienten)

$$a_j^{(n+1)} = a_j^{(n)} + \alpha \widehat{d}_n \widehat{x}_{n-j} \text{ bzw. vektoriell}$$

$$\mathbf{A}^{(n+1)} = \mathbf{A}^{(n)} + \alpha \widehat{d}_n \widehat{X}_{n-1} \text{ mit}$$

$$\widehat{X}_n = \begin{bmatrix} \widehat{x}_n \\ \widehat{x}_{n-1} \\ \vdots \\ \widehat{x}_{n-N+1} \end{bmatrix}$$

Delta-Modulierung

Für die Sprachcodierung wird gerne der sog. *Δ-Modulator* verwendet, d. i. ein DPCM-Verfahren mit 1-Bit-Quantisierer.

D.h.: der Codierer dem Decodierer quasi nur mitteilen darf, ob sich der aktuelle Eingabewert im Vergleich zum vorigen erhöht oder erniedrigt hat;
der Decodierer interpretiert dies, indem er $\pm\Delta$ rechnet.
Fehler ?! $\leadsto$ hohe Abtastrate erforderlich.
Dennoch ist bei fixiertem 1-Bit-Quantisierer zu beobachten:

- Ist die Quelle fast konstant, so schwingt der Modulator.
- Bei steilem Signalanstieg oder -abfall kommt er nicht so schnell nach.

Zur **Wahl von Δ**:

- Ist Δ klein, so beobachten wir schwaches Schwingen, aber schlechtes Nacheilen.

- Ist Δ groß, so sind auch die Schwingungen groß, aber das Nacheilverhalten ist gut.

Auch hier bietet sich eine *adaptive Delta-Modulierung* an; wir betrachten im folgenden die sog. *CFDM* (engl.: constant factor delta modulation).

Details zu CFDM

Δ_n: den vom Decodierer erhaltenen vorzeichenbehafteten Differenzterm im n-ten Schritt.
Da die nachstehend angegebene Adaption mit den Faktoren $M > 1$ nicht das Vorzeichen von Δ_n beeinflusst, ist folgende Festlegung nicht zirkulär:

$$s_n = \begin{cases} 1 & \text{wenn } \Delta_n > 0 \\ -1 & \text{wenn } \Delta_n < 0 \end{cases}$$

$$\Delta_n = \begin{cases} M\Delta_{n-1} & \text{wenn } s_n = s_{n-1} \\ M^{-1}\Delta_{n-1} & \text{wenn } s_n \neq s_{n-1} \end{cases}$$

CFDM lässt sich verbessern, indem nicht nur die letzte Ausgabe des Codierers in Betracht gezogen wird.

Beispiele zu CFDM

Man mag sich klarmachen, was die folgenden Fälle „bedeuten”:

$$\begin{aligned} s_n &\neq s_{n-1} = s_{n-2} \\ s_n &= s_{n-1} \neq s_{n-2} \\ s_n &= s_{n-1} = s_{n-2} \end{aligned}$$

Für die Sprachcodierung wird empfohlen, die Formel $\Delta_n = M_i \Delta_{n-1}$ für folgende Werte / Fälle von M_i zu verwenden:

$$\begin{aligned} s_n \neq s_{n-1} = s_{n-2} &\Rightarrow M_1 = 0,4 \\ s_n \neq s_{n-1} \neq s_{n-2} &\Rightarrow M_2 = 0,9 \\ s_n = s_{n-1} \neq s_{n-2} &\Rightarrow M_3 = 1,5 \\ s_n = s_{n-1} = s_{n-2} &\Rightarrow M_4 = 2,0 \end{aligned}$$