

Lehrveranstaltung

Informationstheorie

— Sommersemester 2023 —

Martin Mittelbach (Vorlesung, Tutorium), Anne Wolf (Übung, Tutorium)
{martin.mittelbach, anne.wolf}@tu-dresden.de

Professur für Informationstheorie und maschinelles Lernen, TU Dresden

*Vorlesung 2
12. April 2023*

Inhalt der ersten Vorlesung

- Organisatorisches, Einführung, Motivation, Themenübersicht
- Wiederholung Grundbegriffe diskrete W-Theorie, Notation
 ⇒ Fortsetzung gleich
- Online-Umfrage zur Komprimierbarkeit zweier Bilddateien
 ⇒ Auswertung gleich

Wiederholung Grundbegriffe W-Theorie

Übersicht:

- Diskrete Zufallsgröße X ✓, W-Funktion p_X ✓, Alphabet \mathcal{X} ✓,
Träger \mathcal{S}_X ✓, transformierte Zufallsgröße $g(X)$ ✓,
Erwartungswert $\mathbb{E}(X), \mathbb{E}(g(X))$ ✓, Varianz $\text{var}(X)$ ✓
 - Schreibweisen
 - Gleichverteilung ✓, identisch verteilte Zufallsgrößen ✓
 - Diskreter Zufallsvektor (X, Y) ✓, gemeinsame W-Funktion $p_{X,Y}$ ✓,
Rand-W-Funktion p_X, p_Y ✓, Unabhängigkeit ✓, Kovarianz ✓,
Unkorreliertheit ✓
-

Fortsetzung mit Folien der Vorlesung 1 (Folien 34–37):

- Bedingte W-Funktion, bedingte Rand-W-Funktion, bedingte
Unabhängigkeit, Markowkette der Länge $3, 4, \dots, n$

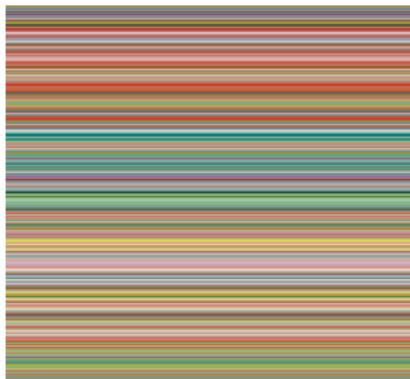
Umfrage zur Komprimierbarkeit aus Vorlesung 1

Die Frage lautete (Vorlesung 1, Folie 38):

Was denken Sie, welche der beiden nachfolgenden Bilddateien (Pixelanzahl, Farbformat, Farbtiefe identisch) lässt sich mit verlustloser Datenkompression stärker komprimieren?

- A) Das Bild mit dem Titel: "Strip (927-9)"
- B) Das Bild mit dem Titel: "Abstraktes Bild (946-3)"

(Unkomprimiert: 1.170.000 Pixel \times 3 Farikanäle \times 8 Bit Farbtiefe = 28.080.000 Bit = 3.510.000 Byte)



Gerhard Richter: Strip (927-9), 2012



Gerhard Richter: Abstraktes Bild (946-3), 2016

Umfrage zur Komprimierbarkeit aus Vorlesung 1

Ergebnis der Umfrage: 15 Abstimmungen (Danke!), dabei stimmten

- 100 % für das Bild A mit dem Titel "Strip (927-9)"
- 0 % für das Bild B mit dem Titel "Abstraktes Bild (946-3)"

Abgegebene Kommentare: (tlw. aus SoSe 2022)

Bild 2 ist "zufälliger" als Bild 1, deswegen kann durch eine Prädiktion nicht so stark komprimiert werden.

Es lässt sich bei dem Bild "Strip (927-9)" jeder Pixelzeile ein Farbwert zuordnen, während bei dem Bild "Abstraktes Bild (946-3)" jedem Pixel einzeln ein Farbwert zugeordnet werden muss, da es **keine wiederkehrenden Muster** in dem Bild gibt. Dadurch kann man mit weniger Informationen das erste Bild verlustlos rekonstruieren.

Das Bild ist **mehr regulär**, deshalb **weniger Informationen**. Es liegt größere Möglichkeit, mit Informationen von Pixels in der Nähe das Bild wiederzuherstellen.

Weil die Streifen ein **wiederkehrendes Muster** sind, das mit weniger Daten ausgedrückt werden kann.

"Strip (927-9)" kann vektorisiert werden, da die Bildpunkte auf der gleichen Zeile die gleiche Farben haben und man somit **Eigenschaften zeilenweise zusammenfassen** kann. Bei "Abstraktes Bild (946-3)" kann man Eigenschaften schlechter zusammenfassen.

Das Bild Strip lässt sich **stärker komprimieren**, da lediglich eine Spalte an Pixeln (3120Byte) gespeichert werden muss + die Regel, dass die Pixel jeder Reihe den selben Farbwert haben. Beim anderen Bild gibt es **keine so eindeutigen Strukturen**, sodass dieses Bild sich sicherlich **weniger stark komprimieren** lässt.

Im Prinzip müsste man nur die erste "Pixelspalte" speichern und auf wie viele Pixel dieses breitgezogen werden muss.

Weil bei "Strip" **Flächen gleicher Farbe** streifenweise zusammengefasst werden können. Bei "Abstraktes Bild" ist nicht so eine einfache Abhängigkeit zwischen Position und Farbe des Pixels zu finden.

Umfrage zur Komprimierbarkeit aus Vorlesung 1

Ergebnis der Umfrage: 15 Abstimmungen (Danke!), dabei stimmten

- 100 % für das Bild A mit dem Titel "Strip (927-9)"
- 0 % für das Bild B mit dem Titel "Abstraktes Bild (946-3)"

Abgegebene Kommentare: (tlw. aus SoSe 2022)

Möglicherweise kann man hier Bits je Farbkanal einsparen, da nur wenige Farbtöne vorhanden sind (keine Farbverläufe wie beim anderen Bild). Ferner könnte man zusammenhängende Bereiche gleicher Farbe statt einzelner Pixel beschreiben. Da die Kompression verlustfrei erfolgen soll, wäre die Gruppierung gleicher Farbpixel beim zweiten Bild ("Abstraktes Bild (946-3)") nur durch sehr granulare Darstellung realisierbar, was letztendlich einen höheren Speicherbedarf bedeuten würde.

Das Bild "Strip" wirkt insgesamt simpler als das Bild "Abstraktes Bild". Es würde keine Informationen über jedes Pixel brauchen, um das Bild zu rekonstruieren. Damit sollte es sich auch stärker komprimieren lassen.

...

Verlustlose Kompression im PNG-Format:

(Speicherung mit Adobe Photoshop Version 20.0.6, PNG-Format-Option: "Kleinste Dateigröße")

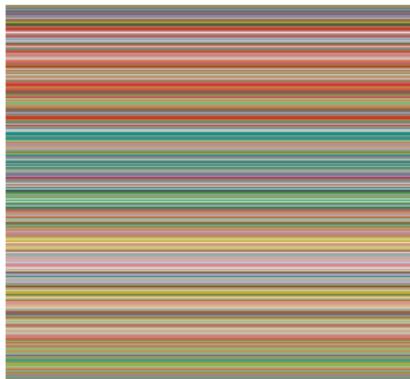
- Bild A mit dem Titel "Strip (927-9)": 10.033 Byte
- Bild B mit dem Titel "Abstraktes Bild (946-3)": 2.878.211 Byte

Umfrage zur Komprimierbarkeit aus Vorlesung 1

Allgemeine Eigenschaften der Bilddateien:

Bild A "Strip (927-9)": regelmäßige Ordnung/Struktur, häufige Wiederholung, eher deterministische Anordnung, "Informationsgehalt" geringer als bei Bild B
später: $\hat{=}$ niedrige Entropie

Bild B "Abstraktes Bild (946-3)": wenig Ordnung/Struktur, kaum Wiederholung, eher zufällige Farbanordnung, "Informationsgehalt" höher als bei Bild A
später: $\hat{=}$ hohe Entropie



Gerhard Richter: Strip (927-9), 2012



Gerhard Richter: Abstraktes Bild (946-3), 2016

Vorgehensweise / Themenübersicht



Vorgehensweise / Themenübersicht



Inhalt Vorlesung 2

- 1. Verlustlose Datenkompression mit Codes variabler Länge
 - (1.1) Einführendes Beispiel, Modellbildung, Problemstellung
 - (1.2) Quellen als Datenmodell
 - (1.3) Codes variabler Länge
-

1. Verlustlose Datenkompression mit Codes variabler Länge

(1.1) Einführendes Beispiel, Modellbildung, Problemstellung

- **Aufgabe:** Es soll eine Datenmenge bestehend aus 10.000 Zeichen mit folgender Zusammensetzung

Zeichen	Häufigkeit	relative Häufigkeit
1	5000	$\frac{1}{2}$
2	2500	$\frac{1}{4}$
3	1250	$\frac{1}{8}$
4	1250	$\frac{1}{8}$

binär abgespeichert werden.

- **Binäre Codierung:**

Variante 1

1	→	00
2	→	01
3	→	10
4	→	11

Variante 2

1	→	0
2	→	10
3	→	110
4	→	111

Kanonische Binärdarstellung

Häufige Zeichen werden durch **kurze Codewörter** und seltene Zeichen durch **lange Codewörter** repräsentiert.

Speicherbedarf:

Variante 1: 20.000 Bit für gesamte Datenmenge bzw. 2 Bit pro Zeichen

Variante 2: 17.500 Bit für gesamte Datenmenge bzw. 1.75 Bit pro Zeichen im Mittel

(1.1) Einführendes Beispiel, Modellbildung, Problemstellung

- **Modellbildung:**

- Wir wollen **Daten** als Datenstrom auffassen, d. h. **als Sequenz von Symbolen**/ Zeichen eines bestimmten (endlichen) Wertevorrats, d. h. "Alphabets".
 - **Textdaten:** Sequenz von Buchstaben und (Satz-)Zeichen
 - **Audiodata:** Sequenz von Tönen
 - **Bilddaten:** Sequenz von Farbwerten (2-dimensional angeordnet)
 - **Videodata:** Sequenz von Bild- und Audiodata
- Die **Symbole** (Buchstaben, Töne, Farbwerte) dieser Sequenzen werden jeweils digital, d. h. **durch Bits repräsentiert**.
- **Relevant** für die Datenkompression sind **Häufigkeiten** verwendeter Symbole und Symbolketten sowie **Abhängigkeiten der Symbole** untereinander.
- Daraus resultiert der **Ansatz**, Daten durch ein **stochastisches Modell** zu repräsentieren.
- Solch ein abstrahiertes **mathematisches Modell** bildet nicht die **Realität**, d. h. die **realen Daten**, **ab**, sondern lediglich die **wesentlichen Aspekte**, die für die Analyse der Datenkompression relevant sind.

(1.1) Einführendes Beispiel, Modellbildung, Problemstellung

- **Mathematisches Modell:** Die Erzeugung von Daten wird durch Zufallsexperimente modelliert, welche wiederum durch Zufallsgrößen beschrieben werden.

Menge der möglichen Symbole	\triangleq	Alphabet einer Zufallsgröße
relative Häufigkeit eines Symbols	\triangleq	W für Ereignis, dass Zufallsgröße = Symbol (ausreichend große Stichprobe vorausgesetzt)
Erzeugung eines Symbols	\triangleq	einmaliges Zufallsexperiment, durch eine Zufallsgröße beschrieben
Erzeugung einer Symbolfolge, d. h. eines Datenstroms	\triangleq	mehrere Zufallsexperimente, durch Folge von Zufallsgrößen beschrieben

- **Mathematisches Modell für einführendes Beispiel:** Folge von Zufallsgrößen

$$X_1, X_2, X_3, X_4 \dots$$

wobei jedes Folgeglied X_k das gleiche Alphabet $\mathcal{X} = \{1, 2, 3, 4\}$ und die gleiche W-Funktion p mit folgenden Werten besitzt.

x	1	2	3	4
$p(x)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$

(1.1) Einführendes Beispiel, Modellbildung, Problemstellung

- **Codierung eines Datenstroms:** Ein Datenstrom entspricht einer konkreten Folge von Werten der Zufallsgrößen. Beispielsweise könnten ein **Datenstrom** und dessen **binäre Codierung gemäß Variante 2 für obiges Beispiel** wie folgt aussehen.

Folge von Zufallsgrößen:	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	...
mögliche Werte der Zufallsgrößen:	1	2	1	1	3	2	1	4	...
	↓	↓	↓	↓	↓	↓	↓	↓	...
binäre Codierung (Variante 2):	0	10	0	0	110	10	0	111	...

- Die gewählte Codierung hat folgende Vorzüge:
 - Jede endliche Aneinanderreihung von Codewörtern ist ohne Trennzeichen rekonstruierbar.
 - Kein Codewort ist Anfang eines anderen Codewortes, d. h. das Ende eines Codewortes kann beim "Abschreiten" der Bitfolge unmittelbar erkannt werden.

(1.1) Einführendes Beispiel, Modellbildung, Problemstellung

- **Problemstellung:** Für die in der Variante 2 gewählte Codierung benötigt man in obigem Beispiel im Mittel 1.75 Bit pro Symbol.
 - Dieser Wert ist identisch zu dem in Übung 1, Aufgabe 1d) betrachteten Erwartungswert

$$\mathbb{E}(-\log_2(p(X_k))) = \sum_{x \in \mathcal{X}} -\log_2(p(x)) \cdot p(x),$$

für die in diesem Beispiel betrachtete W-Funktion.

- Ist das Zufall?
- Gibt es einen Code für dieses Beispiel, der die gleichen Eigenschaften wie der in Variante 2 hat, aber im Mittel weniger Bit pro Zeichen benötigt?
- **Zielstellung:** In diesem Abschnitt betrachten wir die verlustlose Quellencodierung, d.h. es geht um eine möglichst effiziente Repräsentation von Daten mittels Codes variabler Länge zum Zweck der Speicherung. Die Effizienz einer Codierung werden wir mit Hilfe der sogenannten mittleren Codewortlänge beurteilen.

(1.2) Quellen als Datenmodell

- **Datenmodell:** Wir verwenden als **Datenmodell** eine
Folge von diskreten Zufallsgrößen

und nennen dieses Modell

(diskrete) Quelle

(auch: Daten~, Informations~).

- **Folgen von diskreten Zufallsgrößen:**

- In der Wiederholung in Vorlesung 1 haben wir diskrete Zufallsgrößen und Zufallsvektoren (= zwei oder endlich viele diskrete Zufallsgrößen) betrachtet.
- Eine **Folge von diskreten Zufallsgrößen** (auch: zufällige Folge) ist eine natürliche Erweiterung und besteht aus **abzählbar unendlich vielen**, linear geordneten, **diskreten Zufallsgrößen**.
- **Schreibweisen:**

$$X = (X_1, X_2, X_3, \dots)$$

oder

$$X = (X_k)_{k \in \mathbb{N}} \quad \text{oder} \quad X = (X_k, k \in \mathbb{N})$$

($\mathbb{N} = \{1, 2, 3, \dots\}$ = Menge der natürlichen Zahlen)

(1.2) Quellen als Datenmodell

- **Folgen von diskreten Zufallsgrößen:** [...]

- Vereinbarung im Folgenden: Das **Alphabet** ist für alle Folgeglieder **identisch** ($= \mathcal{X}$) und **endlich**. Es wird **Alphabet der Folge** bzw. **Alphabet der Quelle** genannt.
- Für die **eindeutige Festlegung** der **W-Verteilung** einer Folge $(X_k)_{k \in \mathbb{N}}$ von diskreten Zufallsgrößen benötigt man die **W-Funktionen** p_{X_1, X_2, \dots, X_n} mit

$$p_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) = \mathbb{P}(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n), \\ x_1, x_2, \dots, x_n \in \mathcal{X}$$

der n -dimensionalen diskreten Zufallsvektoren

$$(X_1, X_2, \dots, X_n)$$

für alle $n \in \mathbb{N}$.

$$\left(\underbrace{X_1}_{\sim p_{X_1}}, X_2, X_3, \dots, X_n, X_{n+1}, X_{n+2}, \dots \right)$$
$$\underbrace{\quad}_{\sim p_{X_1, X_2}}$$
$$\underbrace{\quad}_{\sim p_{X_1, X_2, X_3}}$$
$$\underbrace{\quad}_{\sim p_{X_1, X_2, \dots, X_n}}$$
$$\underbrace{\quad}_{\sim p_{X_1, X_2, \dots, X_{n+1}}}$$

(1.2) Quellen als Datenmodell

- **Stationäre Folge diskreter Zufallsgrößen:** Eine Folge $(X_k)_{k \in \mathbb{N}}$ diskreter Zufallsgrößen heißt

stationär

(auch: streng stationär, im engeren Sinne stationär), falls für alle $m, n \in \mathbb{N}$ und $k_1 < k_2 < \dots < k_n \in \mathbb{N}$ gilt:

$$(X_{k_1}, X_{k_2}, \dots, X_{k_n}) \sim (X_{k_1+m}, X_{k_2+m}, \dots, X_{k_n+m}).$$

(Erinnerung: \sim heißt "identisch verteilt")

Das bedeutet, die W-Verteilungen sämtlicher endlicher Teilabschnitte der Folge $(X_k)_{k \in \mathbb{N}}$ sind invariant gegenüber (zeitlichen) Verschiebungen. Speziell gilt

$$X_1 \sim X_2 \sim X_3 \sim X_4 \sim \dots \sim \dots$$

und für $k = 1, 2, 3, \dots$

$$(X_1, X_{1+k}) \sim (X_2, X_{2+k}) \sim (X_3, X_{3+k}) \sim (X_4, X_{4+k}) \sim \dots \sim \dots$$

usw.

(1.2) Quellen als Datenmodell

- **Unabhängige Folge diskreter Zufallsgrößen:** Eine Folge $(X_k)_{k \in \mathbb{N}}$ diskreter Zufallsgrößen heißt

unabhängig

falls für alle $n \in \mathbb{N}$ die Komponenten des diskreten Zufallsvektors (X_1, X_2, \dots, X_n) (stochastisch) unabhängig sind, d. h. wenn gilt

$$p_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) = p_{X_1}(x_1) \cdot p_{X_2}(x_2) \cdot \dots \cdot p_{X_n}(x_n), \\ x_1, x_2, \dots, x_n \in \mathcal{X}.$$

- **i.i.d.-Folge diskreter Zufallsgrößen:** Eine unabhängige Folge $(X_k)_{k \in \mathbb{N}}$ diskreter Zufallsgrößen heißt

i.i.d.-Folge

falls zusätzlich sämtliche Folgeglieder identisch verteilt sind (i.i.d. = independent and identically distributed).

- Eine i.i.d.-Folge diskreter Zufallsgrößen ist stationär.
- Sie bildet die einfachste Form einer Folge von Zufallsgrößen.
- Ihre W-Verteilung ist eindeutig durch die W-Funktion p_{X_1} des ersten Folgegliedes festgelegt.

(1.2) Quellen als Datenmodell

- **Markowkette diskreter Zufallsgrößen:** Eine Folge $(X_k)_{k \in \mathbb{N}}$ diskreter Zufallsgrößen heißt

Markowkette (erster Ordnung)

falls für alle $n \in \{3, 4, 5, \dots\}$ gilt (siehe Vorlesung 1, Folien 36/37)

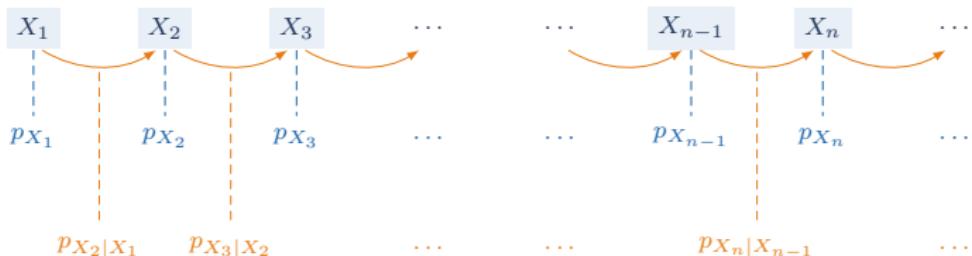
$$(X_1, X_2, \dots, X_{n-2}) \rightarrow X_{n-1} \rightarrow X_n.$$

(Markowkette m -ter Ordnung \rightarrow Skript (§0.77))

- Die Definition ist äquivalent dazu, dass für alle $n \in \{3, 4, 5, \dots\}$,
 $(x_{n-1}, x_{n-2}, \dots, x_1) \in \mathcal{S}_{(X_{n-1}, X_{n-2}, \dots, X_1)}$ und $x_n \in \mathcal{X}$ gilt

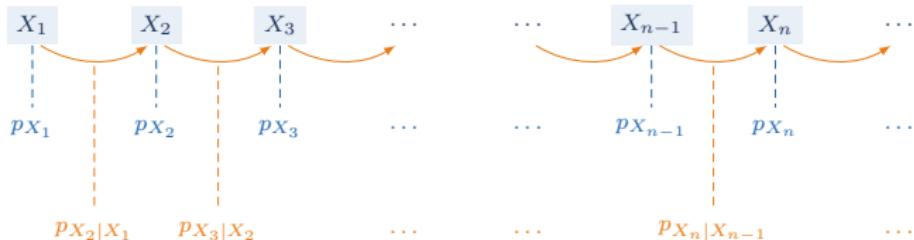
$$p_{X_n|(X_{n-1}, (X_{n-2}, \dots, X_1))}(x_n | (x_{n-1}, (x_{n-2}, \dots, x_1))) = p_{X_n|X_{n-1}}(x_n | x_{n-1}).$$

(Die **violettfarbenen** Klammern werden üblicherweise weggelassen.)



(1.2) Quellen als Datenmodell

- Markowkette diskreter Zufallsgrößen:

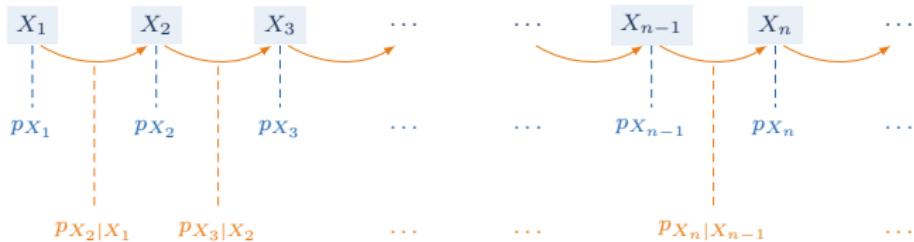


- Die W-Verteilung einer Markowkette diskreter Zufallsgrößen ist durch die W-Funktion p_{X_1} des ersten Folgegliedes (Anfangsverteilung) und die bedingten W-Funktionen $p_{X_k|X_{k-1}}$ für alle $k \in \{2, 3, 4, \dots\}$ eindeutig festgelegt, denn es gilt für alle $n \in \{2, 3, 4, \dots\}$

$$\begin{aligned} p_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) \\ = p_{X_1}(x_1) \cdot p_{X_2|X_1}(x_2|x_1) \cdot \dots \cdot p_{X_n|X_{n-1}}(x_n|x_{n-1}), \\ x_1, x_2, \dots, x_n \in \mathcal{X}. \end{aligned}$$

(1.2) Quellen als Datenmodell

- Markowkette diskreter Zufallsgrößen:



- Eine **Markowkette** diskreter Zufallsgrößen heißt

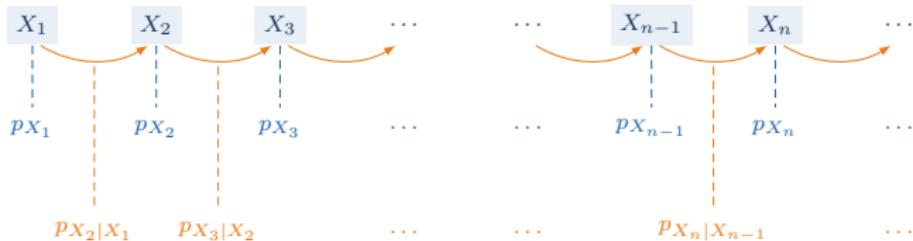
(zeitlich) homogen

falls die bedingte W-Funktion $p_{X_k|X_{k-1}}$ von X_k unter der Bedingung X_{k-1} für alle $k \in \{2, 3, 4, \dots\}$ identisch, d.h. **(zeitlich) invariant** ist.

- Die W-Verteilung einer homogenen Markowkette diskreter Zufallsgrößen ist durch die W-Funktion p_{X_1} und die bedingte W-Funktion $p_{X_2|X_1}$ eindeutig festgelegt.

(1.2) Quellen als Datenmodell

- **Markowkette diskreter Zufallsgrößen:**



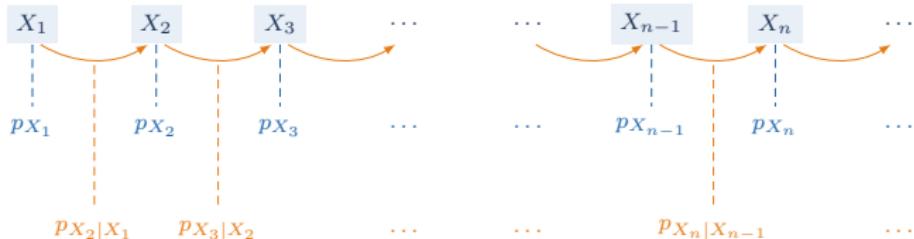
- Eine **homogene Markowkette** diskreter Zufallsgrößen, für die zusätzlich sämtliche Folgeglieder identisch verteilt sind, d. h. $X_1 \sim X_2 \sim X_3 \sim \dots \sim \dots$, ist stationär und heißt

stationäre Markowkette.

- Eine **stationäre Markowkette** bildet nach der i.i.d.-Folge in gewisser Hinsicht die "zweiteinfachste" Folge von Zufallsgrößen, bei der spezielle Abhängigkeiten zwischen Folgegliedern berücksichtigt werden, d. h. eine Art Gedächtnis.

(1.2) Quellen als Datenmodell

- Markowkette diskreter Zufallsgrößen:

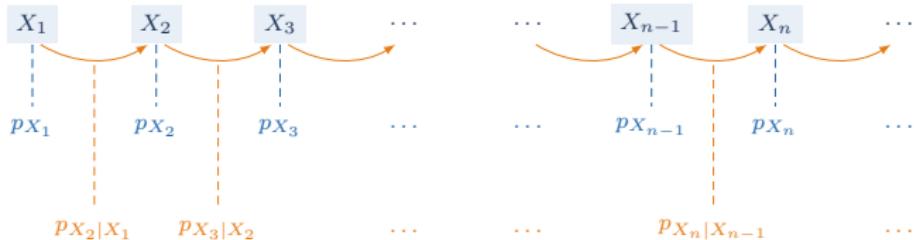


- Das für alle Folgeglieder identische Alphabet \mathcal{X} einer Markowkette wird auch als **Zustandsraum** bezeichnet und die Elemente von \mathcal{X} als Zustände.
- Die durch $p_{X_k|X_{k-1}}$ gegebenen bedingten Wen für aufeinanderfolgende Folgeglieder nennt man auch **Übergangs-Wen**. Sie werden oft in Matrixform angegeben. (Annahme für Alphabet hier $\mathcal{X} = \{a_1, a_2, \dots, a_m\}$)

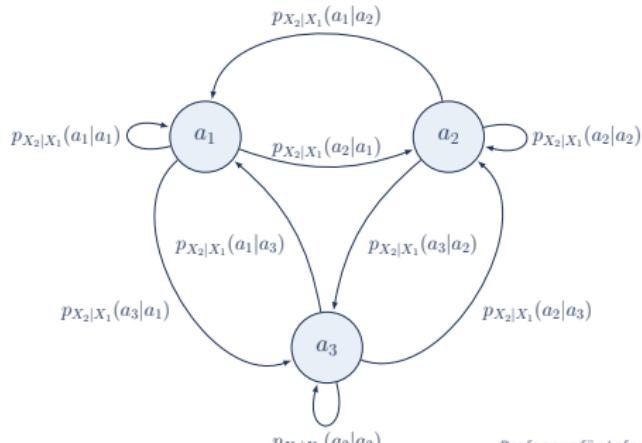
$$\begin{pmatrix} p_{X_k|X_{k-1}}(a_1|a_1) & p_{X_k|X_{k-1}}(a_2|a_1) & \dots & p_{X_k|X_{k-1}}(a_m|a_1) \\ p_{X_k|X_{k-1}}(a_1|a_2) & p_{X_k|X_{k-1}}(a_2|a_2) & \dots & p_{X_k|X_{k-1}}(a_m|a_2) \\ \dots & \dots & \dots & \dots \\ p_{X_k|X_{k-1}}(a_1|a_m) & p_{X_k|X_{k-1}}(a_2|a_m) & \dots & p_{X_k|X_{k-1}}(a_m|a_m) \end{pmatrix}$$

(1.2) Quellen als Datenmodell

- Markowkette diskreter Zufallsgrößen:



- Für eine homogene Markowkette wird die Matrix der Übergangs-Wen oft als **Zustandsgraph** illustriert. (Illustration für Alphabet $\mathcal{X} = \{a_1, a_2, a_3\}$)



(1.2) Quellen als Datenmodell

- Bezeichnungen für spezielle Quellen:

Informationstheorie	Wahrscheinlichkeitstheorie
• (diskrete) Quelle	= Folge diskreter Zufallsgrößen
• (diskrete) stationäre Quelle	= stationäre Folge diskreter Zufallsgrößen
• (diskrete) gedächtnislose Quelle	= unabhängige Folge diskreter Zufallsgrößen
• (diskrete) stationäre gedächtnislose Quelle	= i.i.d.-Folge diskreter Zufallsgrößen
• (diskrete) Markow-Quelle	= Markowkette diskreter Zufallsgrößen
• (diskrete) stationäre Markow-Quelle	= stationäre Markowkette diskreter Zufallsgrößen

- Bemerkungen:

- Da wir nur diskrete Quellen betrachten, lassen wir den Zusatz "diskret" oft weg.
- Achtung: Im Skript wird für diskrete stationäre gedächtnislose Quellen die Klassifizierung "stationär" weggelassen, da nur stationäre Quellen betrachtet werden. Dies ist auch eine verbreitete Konvention in der Literatur (häufige Abkürzung: DMS = "discrete memoryless source").