Aufgabenstellung

- Muster- / Strukturerkennung
- Vorhersage / Abschätzung der Entwicklung des Systems / der Umgebung in naher/ ferner Zukunft

Was ist notwendig?

- Ein Agent (z.B. Physiker)
- Möglichkeit Daten aufzunehmen (z.B. Messgeräte)
- Eine Datenbank, in der die Muster abgespeichert sind, die wir erkennen können (z.B. Erfahrung)

Übersetzung

- Struktur= Funktion
- Daten=Graph dieser Funktion
- Musterbibliothek=Schar von Funktionen
- Struktur erkennen=passende Funktion aus dieser Schar auswählen

<u>Bemerkung:</u> wir beschränken uns auf Abb (R,R)

Kontrolle der Auswahl

- Normen bzw. Metriken
- Annahme : f,g aus L^2

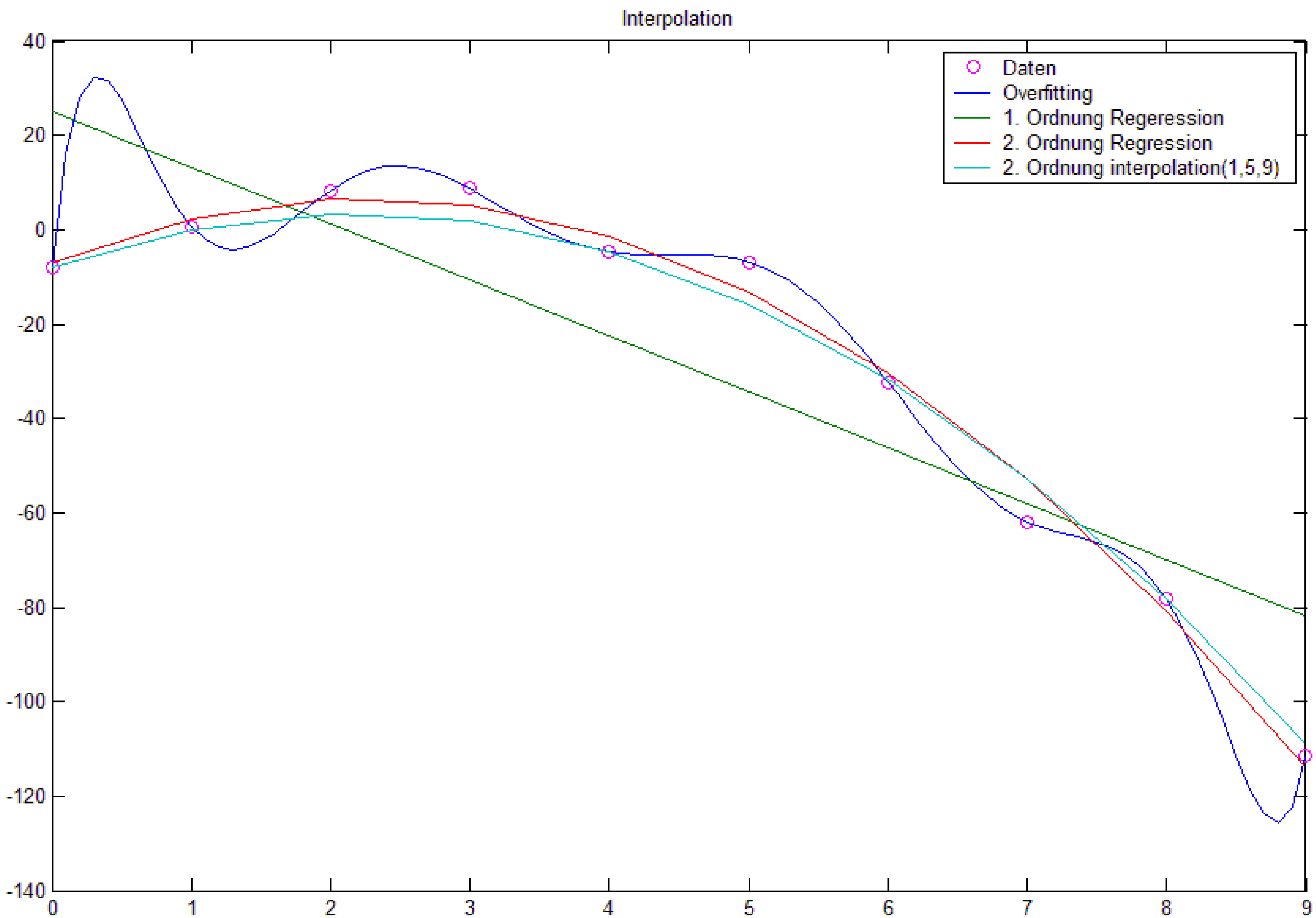
-> $d:=S\ (f-g)^2 dt$ – eine Metrik

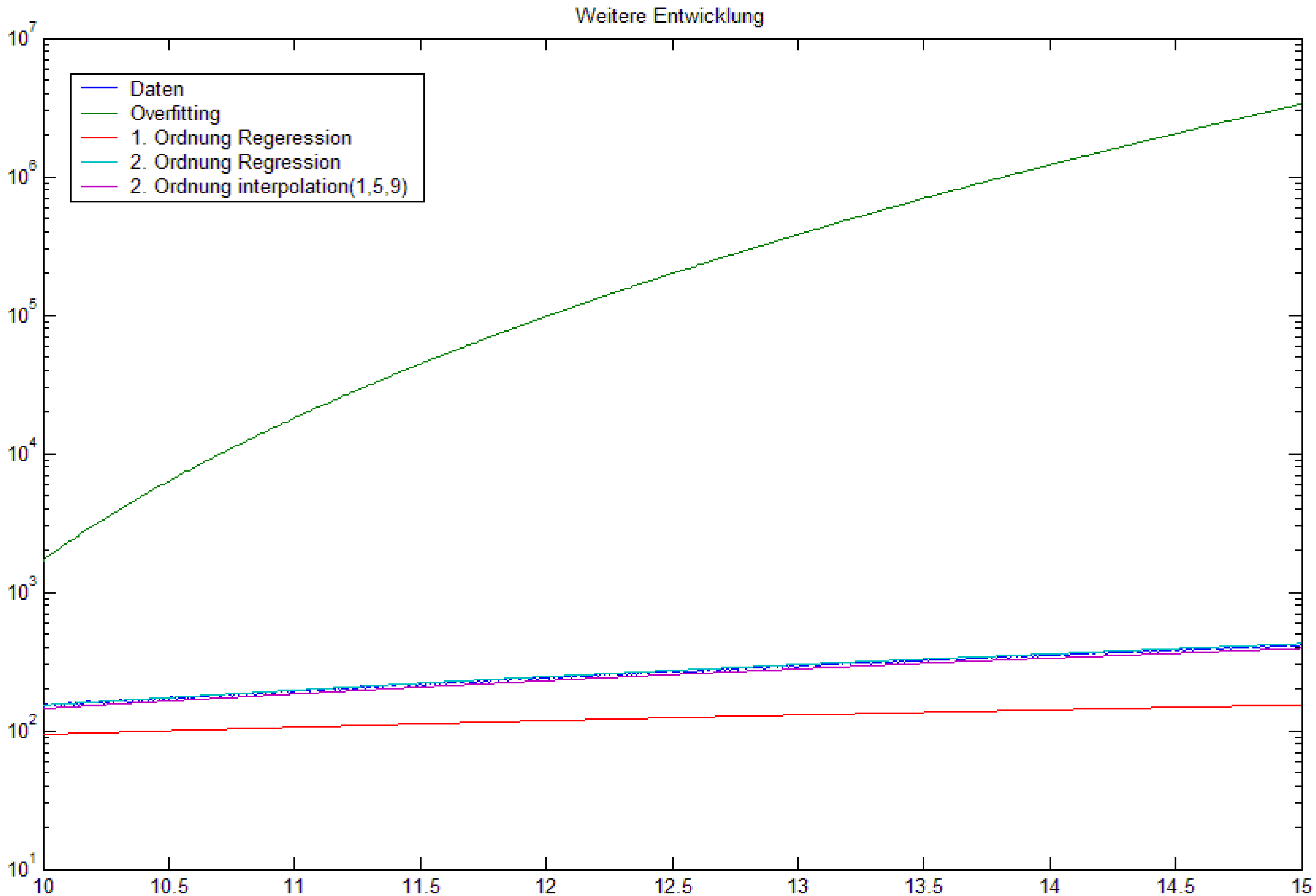
Daten sind aber nicht kontinuierlich

-> d muss abgeschätzt werden

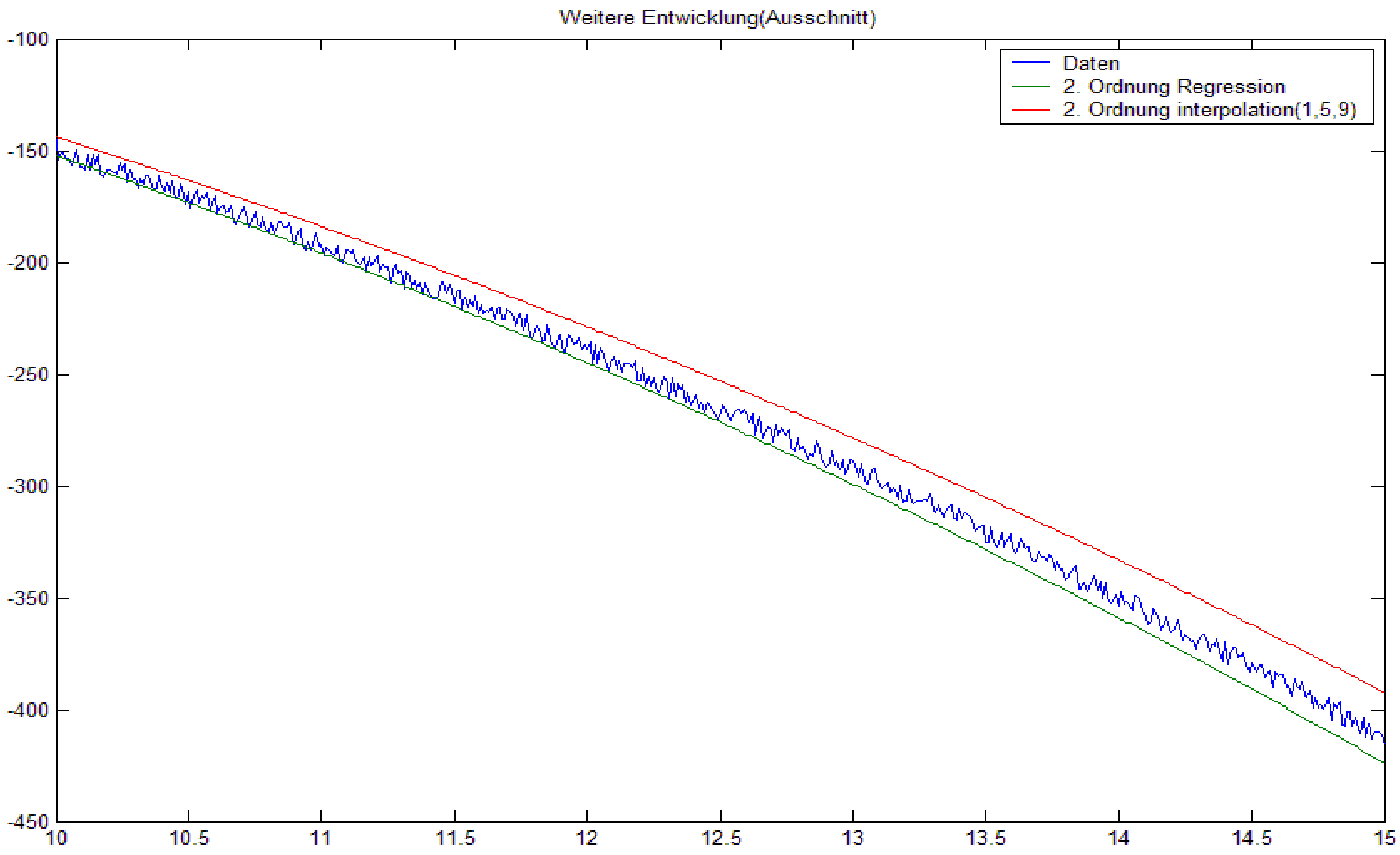
$d_a := \Sigma\ (f(t_i)-g(t_i))^2$

Problem:: Overfitting


Interpolation
Daten
Overfitting
1. Ordnung Regeression
2. Ordnung Regression
2. Ordnung interpolation(1,5,9)
40
20
0
-20
-40
-60
-80
-100
-120
-140
0
1
2
3
4
5
6
7
8
9



Weitere Entwicklung
Daten
Overfitting
1. Ordnung Regeression
2. Ordnung Regression
2. Ordnung interpolation(1,5,9)
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10
10.5
11
11.5
12
12.5
13
13.5
14
14.5
15



Weitere Entwicklung(Ausschnitt)
Daten
2. Ordnung Regression
2. Ordnung interpolation(1,5,9)
-100
-150
-200
-250
-300
-350
-400
-450
10
10.5
11
11.5
12
12.5
13
13.5
14
14.5
15

Zusammenfassung

- Zu einfach – schlecht
- zu kompliziert -schlecht

-> goldene Mitte suchen

Fazit: ein neues Konzept notwendig

Stochastische Komplexität

1984 von Rissanen vorgeschlagen, und im Zusammenhang mit dem MDL (Minimum Description Length) – Prinzip entwickelt

Idee

- Kompression = Lernen = Vorhersagen
- höhere Kompressionsrate -> mehr gelernt -> genauere Vorhersagen
- Beispiel:

x :	1	2	3	4	5
y:	1	4	9	16	?

Kompression: x-Werte und y=x^2

Was ist jetzt unser Kriterium?

Das MDL – Prinzip!

Gegeben:

verschiedene Modelle mit verschiedenen Kompressionsraten.

Man wähle das Modell mit der höchsten Kompressionsrate.

Wie komprimiert man Daten? (I)

1. Kodierung

   Prefix-Kodierung

   Ergebnis: Folge aus 1s und 0s

2. Man wählt den bestmöglichen Kode

   z.B. A={1,2,3}

   klug: 1=”0”, 2=”10”, 3=”11”

   weniger klug: 1=”001” 2=”010” 3=”100”

Wie komprimiert man Daten? (II)

Große Zahlen speichern -> mehr Platz notwendig

Beispiel:

A={0,1} pro Zahl 1 bit notwendig

B={0...1023} pro Zahl10 bit notwendig

Allgemein:

A={a,a..,a+1,a+1...,a+n-1}

die Länge des Kodes pro Element im Wesentlichen von n abhängig

Wie komprimiert man Daten? (III)

Erstrebenswert: n möglichts klein zu halten.

Zu jedem n eigener Kode

Problem:: Mitteilen welcher Kode benutzt wird

= Die Nummer des Kodes mitteilen.

Anforderung:

Mit dem wachsenden n darf die Länge dieser Mitteilung nicht explodieren.

Beispiel(I)

Voraussetzungen:

Der Agent kennt nur Polynome mit grad(p)<=n.

Die Daten (D) besitzen endliche Genauigkeit

(y hat abzählbar viele Zustände)

Aufgabe:

D zu komprimieren bzw. eine Struktur in D zu finden

Beispiel(II)

Was hat unser Agent zu tun?

- X-Werte kodieren
- das gewählte Polynom p kodieren. (Länge L_p)
- die Werte von y-p(x) kodieren. (Länge $L_{y|p}$)
- versuchen $L_p + L_{y|p}$ zu minimieren

Beispiel(III)

Ist das Problem des Overfittings gelöst?

Im Prinzip: Ja!

Für das Polynom p mit grad (p)=n-1:

$L_{y|p}=0$

$L_p=(n-1)*C'$

-> $L=L_p+L_{y|p}=(n-1)*C$

y-Werte (p=0):

$L=L_{y|0}=n*C'$

Definition

SK(D | M):= Stochastische Komplexität des Datensatzes D bezüglich des Modells M

$$\mathbf{SK(D\,|M)=L_{D|M}}$$

Konkretes triviales Beispiel

Voraussetzungen:

Bekannt: Polynome

Daten:

x :	1	2	3	4	5
y:	1	4	9	16	25

Lösung:

Nach MDL-Prinzip wird $y=x^2$ gewählt.

(weder $y=ax$ noch $y=p$ mit $grad(p)>=5$)

Philosophie des MDL-Prinzips

Annahmen:

(i) Der Agent kennt die Funktionenschar H

(ii) Y=f(X)+S; wobei f aus H und S – eine (probalistische) Störung.

Dann kann man mit Hilfe des MDL -Prinzips f finden.

<u>Wichtig:</u> ob (ii) erfüllt ist total irrelevant

Zweites Beispiel

Sei S – Störung nach Gaußverteilung mit konstanter Varianz.

Rechnung: ......

MDL besagt:

$$\Sigma \; (f(x_i)-y(x_i))^2 \text{ ist zu minimieren.}$$

Also begründet das MDL Prinzip die Regression, die in der Praxis sehr gute Ergebnisse bringt

Probleme

Man kann nur die bekannten Muster unterscheiden

z.B. Die Zahl Pi wird nicht erkannt.

-> Innovationen notwendig

Numerische Verfahren nicht entwickelt/ (Zeit) Aufwand in die Betrachtung nicht einbezogen