

はじめに
プログラム
まとめ

研究発表 **GMDH**

中市新太

富山県立大学
u020025@st.pu-toyama.ac.jp

July 7, 2023

今週取り組んだこと

2/8

GMDH

GMDHについての論文を読み、理解を深める。

GMDH のプログラム

Python のモジュールである GMDHPy の解読、改造をする

前提条件

3/8

実験データ

X_1 から X_{15} までそれぞれ 300 データ存在する
それらから予測する Y データは X_1 から X_5 までしか使用しない

はじめに
プログラム
まとめ

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	y
1	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_10	x_11	x_12	x_13	x_14	x_15		
2	0.37	0.5054	0.7106	0.1616	0.9076	0.9798	0.5471	0.7375	0.2835	0.0515	0.7539	0.8549	0.3732	0.0506	0.2447	1.9303	
3	0.1129	0.955	0.8035	0.6435	0.4376	0.2078	0.5556	0.8149	0.0278	0.5459	0.6247	0.0166	0.3511	0.0741	0.4028	2.0561	
4	0.1136	0.8886	0.2745	0.6553	0.9654	0.615	0.2479	0.4582	0.4874	0.6615	0.7466	0.3737	0.1811	0.0479	0.0612	2.2355	
5	0.0329	0.0094	0.2588	0.0345	0.4008	0.7928	0.7368	0.8873	0.5784	0.8212	0.2407	0.0767	0.0051	0.2512	0.6374	2	
6	0.3805	0.7699	0.7275	0.1517	0.4636	0.0121	0.0305	0.8303	0.7655	0.0278	0.669	0.6843	0.0609	0.7945	0.4013	1.7548	
7	0.7354	0.8672	0.7945	0.9658	0.021	0.3699	0.2567	0.923	0.7832	0.9284	0.7328	0.8333	0.6334	0.4375	0.8107	0.9944	
8	0.2968	0.4029	0.5435	0.7498	0.8562	0.5617	0.1903	0.4007	0.0202	0.5763	0.5035	0.8088	0.9299	0.3654	0.5133	2.2091	
9	0.5724	0.8763	0.9705	0.6604	0.9533	0.275	0.5914	0.2273	0.9677	0.2411	0.5605	0.393	0.4181	0.3229	0.5266	1.5065	
10	0.242	0.9432	0.0029	0.2423	0.8417	0.6887	0.6046	0.9472	0.6514	0.9461	0.2997	0.9198	0.5878	0.7845	0.0909	2.0047	
11	0.4567	0.3742	0.7631	0.4521	0.6591	0.382	0.7396	0.9317	0.9952	0.3235	0.7732	0.6383	0.5585	0.6219	0.7368	2.0004	
12	0.0225	0.3598	0.9265	0.6993	0.8719	0.045	0.3259	0.2647	0.3065	0.0569	0.4556	0.8095	0.3071	0.979	0.8477	2.2562	
13	0.0498	0.8173	0.6264	0.9395	0.138	0.4992	0.1906	0.4107	0.7475	0.346	0.6217	0.0872	0.9325	0.0887	0.6275	2.0223	
14	0.4271	0.7218	0.922	0.2995	0.1007	0.6517	0.1601	0.9765	0.5451	0.0207	0.9994	0.2294	0.1211	0.0112	0.2794	1.6178	
15	0.6766	0.0811	0.6595	0.5498	0.7094	0.3102	0.9638	0.0999	0.5155	0.6716	0.026	0.8356	0.2123	0.5419	0.8328	2.0504	
16	0.2744	0.6185	0.2032	0.8835	0.7982	0.7337	0.7947	0.3358	0.4167	0.6241	0.4986	0.5872	0.9696	0.5496	0.5924	2.1948	
17	0.3205	0.287	0.2072	0.3936	0.1102	0.1373	0.004	0.377	0.6675	0.0126	0.0093	0.2903	0.1137	0.3363	0.299	1.9979	
18	0.4797	0.294	0.2955	0.2213	0.6992	0.4416	0.6221	0.2331	0.4324	0.0462	0.0618	0.1274	0.6257	0.7262	0.6906	2.004	
19	0.68	0.7623	0.6068	0.0032	0.6047	0.2424	0.1682	0.8809	0.6536	0.9133	0.9114	0.3789	0.2794	0.1258	0.5032	1.5378	
20	0.3708	0.4119	0.1833	0.3907	0.4063	0.9599	0.4233	0.5996	0.3647	0.0409	0.6782	0.2208	0.6353	0.1937	0.3724	2.008	
21	0.4686	0.9918	0.7203	0.727	0.2012	0.1905	0.0251	0.4659	0.3043	0.9881	0.5389	0.0278	0.2642	0.3989	0.0483	1.4684	

図 1: 実験データ

200 個までをトレーニングデータ、それ以降をチェックングデータとする。

プログラム

アルゴリズムの流れ

はじめに
プログラム
まとめ

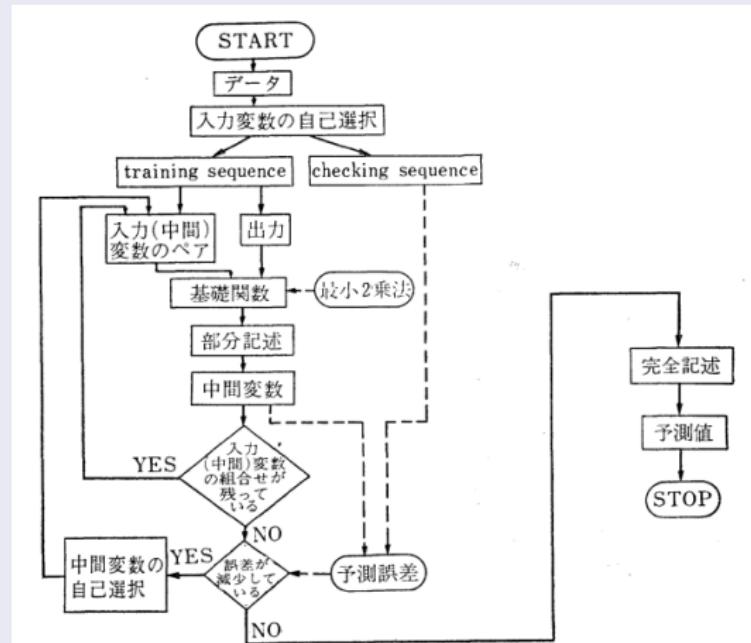


図 2: 流れ

プログラム

答えを導出する式

GMDHPy では答えが 4 式のいずれかから求められる

```
def _transfer_linear(self, u1, u2, w):
    return w[0] + w[1]*u1 + w[2]*u2

def _transfer_linear_cov(self, u1, u2, w):
    return w[0] + u1*(w[1] + w[3]*u2) + w[2]*u2

def _transfer_quadratic(self, u1, u2, w):
    return w[0] + u1*(w[1] + w[3]*u2 + w[4]*u1) + u2*(w[2] + w[5]*u2)

def _transfer_cubic(self, u1, u2, w):
    u1_sq = u1*u1
    u2_sq = u2*u2
    return w[0] + w[1]*u1 + w[2]*u2 + w[3]*u1*u2 + w[4]*u1_sq + w[5]*u2_sq + \
        w[6]*u1*u1_sq + w[7]*u1_sq*u2 + w[8]*u1*u2_sq + w[9]*u2*u2_sq
```

図 3: 4 式

今回は 2 が選択された. (理由不明)

プログラム

6/8

重み

実際に出た重み

頂の値 →
重み
 w_1, w_2 答え

```
min_error:0.0005930788423928066 layer.err: 0.0006115913545489988
[0.10139860025519298 0.7774099893037189 0.3542924847774759 0.7655072918943118
[0.1013986 0.38953071 0.38359476 0.08895633]
1.9957604431484648 1.9956145731055086
1.9986083662306995
[0.101202600075510702 0.76637772710767516 0.241111710112107007 0.7517757085661902]
```

図 4: 方程式

これを前式に当てはめていく

重み w はデータを通じて同じ値だが、 w_1, w_2 は変化していた

結果

7/8

はじめに
プログラム
まとめ

結果

実際の結果. ある程度予測できていることがわかる.

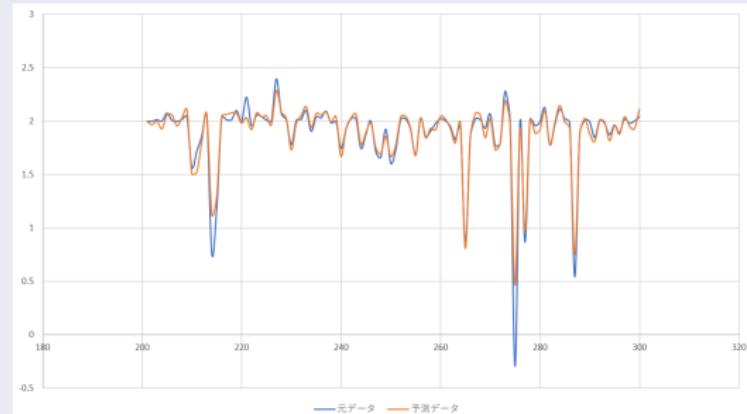


図 5: 結果

まとめ

8/8

今後の展開

- どのように u_1, u_2, w を導出しているのか調べる
- 4式の選択方法を調べる
- ノイズの影響について調べる