Extra

Arthur Charpentier

UQAM

Actuarial Summer School 2019

Self-Promotion: Pricing Games

PolNum	CalYear	Gender	Type	Category	Occupation	Age	Group1	Bonus	Poldur	Value	Adind	SubGroup2	Group2	Density
200285786	2010	Male	E	Large	Employed	48	14	40	0	32345	1	O31	O	35.43401501
200285787	2010	Male	B	Medium	Employed	30	8	-30	9	8995	0	Q29	Q	239.4551701
200285788	2010	Female	B	Large	Housewife	47	2	-50	2	9145	1	U21	U	88.29014956
200285789	2010	Female	D	Large	Self-employed	48	13	-30	15	22075	1	R21	R	275.2822626
200285790	2010	Male	C	Medium	Housewife	57	12	-50	1	24985	1	Q5	Q	99.6400095
200285791	2010	Male	D	Medium	Self-employed	21	15	50	1	12100	1	R11	R	259.0040603
200285792	2010	Male	B	Small	Employed	44	5	-40	15	9820	1	Q10	Q	169.7885554
200285793	2010	Male	F	Small	Self-employed	37	17	-50	5	28680	1	Q5	Q	99.6400095
200285794	2010	Female	C	Large	Retired	49	3	20	4	28470	0	L94	L	84.22903844
200285795	2010	Female	A	Medium	Unemployed	35	5	20	5	8590	0	L112	L	66.06668352
200285796	2010	Male	E	Large	Self-employed	50	10	-30	3	20490	1	Q10	Q	169.7885554
200285797	2010	Female	B	Medium	Housewife	31	8	140	1	8385	1	P28	P	41.2451199
200285798	2010	Female	E	Medium	Self-employed	41	11	90	3	6410	1	L47	L	66.76541883
200285799	2010	Female	A	Medium	Housewife	44	10	-30	8	8485	0	P29	P	20.86448407
200285800	2010	Male	B	Large	Retired	69	8	-40	11	9380	1	U14	U	123.0152076
200285801	2010	Male	F	Medium	Housewife	45	11	30	0	19700	0	L40	L	76.05272599
200285802	2010	Male	E	Large	Retired	53	8	-30	6	10980	1	U19	U	61.79475865
200285803	2010	Male	C	Small	Employed	47	10	-10	9	21980	0	L96	L	45.66982293
200285804	2010	Female	D	Large	Retired	46	7	-50	1	28925	1	U12	U	54.93181221
200285805	2010	Female	C	Large	Retired	67	17	-50	9	14525	1	L52	L	73.25249905

Numtppd	Numtpbi	Indtppd	Indtpbi
0	1	0	1056.0334927
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
3	1	5800.0189068	16.507641942
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

$X_{1,i}$ $X_{k,i}$ Y_i

PolNum	CalYear	Gender	Type	Category	Occupation	Age	Group1	Bonus	Poldur	Value	Adind	SubGroup2	Group2	Density
200375666	2011	Female	A	Large	Employed	46	11	50	0	42975	0	L18	L	58.91132801
200375667	2011	Male	B	Large	Unemployed	31	8	80	11	14835	0	U8	U	125.1320458
200375668	2011	Female	D	Medium	Employed	27	7	-40	13	19000	1	R30	R	296.4319078
200375670	2011	Male	B	Small	Self-employed	22	7	-10	14	33305	0	Q33	Q	129.6690079
200375672	2011	Male	B	Small	Employed	21	17	-20	14	25995	0	T25	T	28.51184808
200375674	2011	Male	C	Medium	Employed	45	19	-50	0	8320	1	N21	N	71.18027901
200375675	2011	Male	C	Medium	Housewife	51	19	30	3	8445	0	L110	L	83.90453994
200375676	2011	Male	E	Large	Self-employed	49	16	-50	3	19545	0	L58	L	64.53563007
200375677	2011	Male	C	Small	Housewife	31	11	-20	5	5030	1	Q7	Q	83.76263662
200375678	2011	Female	A	Medium	Housewife	31	9	-50	14	15480	1	P7	P	25.62227499
200375679	2011	Male	B	Large	Housewife	69	13	-50	7	29580	0	Q23	Q	205.4307964
200375682	2011	Male	A	Medium	Self-employed	43	13	140	3	3735	0	U16	U	91.54176264
200375683	2011	Female	A	Medium	Self-employed	64	18	-20	6	13670	1	O35	O	21.45273029
200375685	2011	Male	E	Large	Employed	25	8	-10	6	17315	0	O22	O	32.18545326
200375688	2011	Male	B	Small	Retired	55	7	-40	3	19410	1	R49	R	208.8164363
200375689	2011	Female	F	Medium	Self-employed	54	9	-40	14	4165	0	U12	U	54.93181221
200375690	2011	Male	D	Large	Housewife	42	9	80	0	11970	1	L125	L	44.16537902
200375692	2011	Male	E	Large	Employed	36	12	-20	7	28415	0	L48	L	71.62174491
200375693	2011	Male	F	Medium	Self-employed	26	10	-30	6	4300	0	L97	L	63.82886936
200375694	2011	Female	B	Small	Unemployed	24	6	-40	7	24005	0	M17	M	201.6569069

Self-Promotion: Pricing Games

Market Rule: insured choose the cheapest premium,

	A	B	C	D	E	F
	787.93	706.97	1032.62	907.64	822.58	**603.83**
	**170.04**	197.81	285.99	212.71	177.87	265.13
	473.15	447.58	**343.64**	410.76	414.23	425.23
	337.98	**336.20**	468.45	339.33	383.55	672.91

Self-Promotion: Pricing Games

Rule: choose randomly among the thre cheapest premium,

	A	B	C	D	E	F
	787.93	706.97	1032.62	907.64	822.58	603.83
	170.04	197.81	285.99	212.71	177.87	265.13
	473.15	447.58	343.64	410.76	414.23	425.23
	337.98	336.20	468.45	339.33	383.55	672.91

RNN (recurrent neural nets)

RNN : Recurrent neural network

Class of neural networks where connections between nodes form a directed graph along a temporal sequence.

Recurrent neural networks are networks with loops, allowing information to persist.

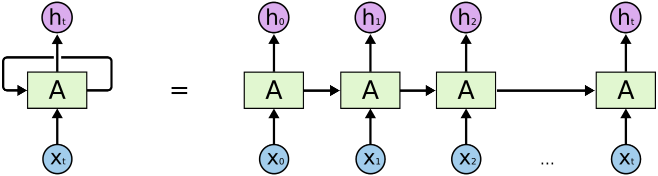
Classical neural net $y_i = m(\boldsymbol{x}_i)$

Recurrent neural net

$$y_t = m(\boldsymbol{x}_t, y_{t-1}) = m(\boldsymbol{x}_t, m(\boldsymbol{x}_{t-1}, y_{t-2})) = \cdots$$

RNN (recurrent neural nets)

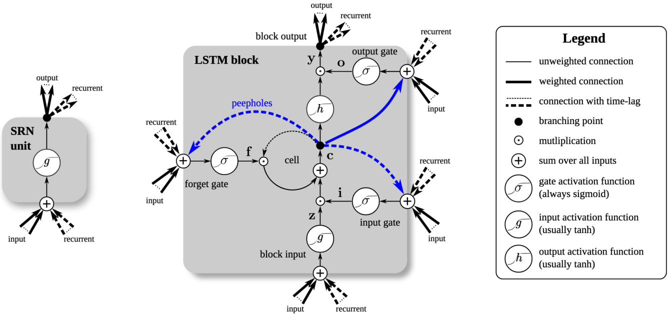
A is the neural net, h is the output (y) and x some covariates.



source https://colah.github.io/

See Sutskever (2017, Training Reccurent Neural Networks)
From recurrent networks to LSTM

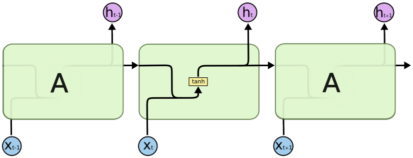
RNN (recurrent neural nets)



source Greff *et al.* (2017, LSTM: A Search Space Odyssey)
see Hochreiter & Schmidhuber (1997, Long Short-Term Memory)

RNN (recurrent neural nets)

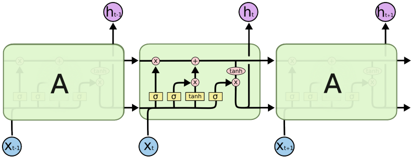
A classical RNN (with a single layer) would be



source https://colah.github.io/

"*In theory, RNNs are absolutely capable of handling such 'long-term dependencies'. A human could carefully pick parameters for them to solve toy problems of this form. Sadly, in practice, RNNs don't seem to be able to learn them*" see Benghio *et al.* (1994, Learning long-term dependencies with gradient descent is difficult)

RNN (recurrent neural nets)



"*RNNs can keep track of arbitrary long-term dependencies in the input sequences.The problem of "vanilla RNNs" is computational (or practical) in nature: when training a vanilla RNN using back-propagation, the gradients which are back-propagated can "vanish" (that is, they can tend to zero) "explode" (that is, they can tend to infinity), because of the computations involved in the process*" (from wikipedia)

RNN (recurrent neural nets)

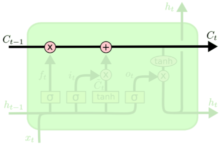
C is the long-term state

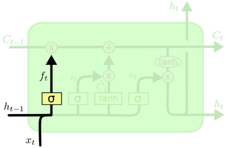
H is the short-term state

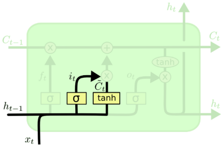
forget gate: $f_t = \text{sigmoid}(\boldsymbol{A}_f[h_{t-1}, x_t] + b_f)$

input gate: $i_t = \text{sigmoid}(\boldsymbol{A}_i[h_{t-1}, x_t] + b_i)$

new memory cell: $\tilde{c}_t = \tanh(\boldsymbol{A}_c[h_{t-1}, x_t] + b_c)$



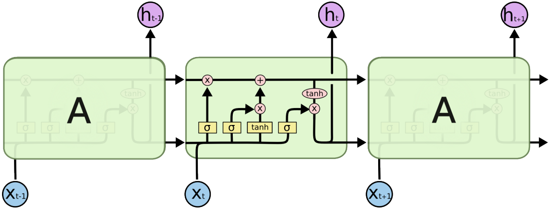


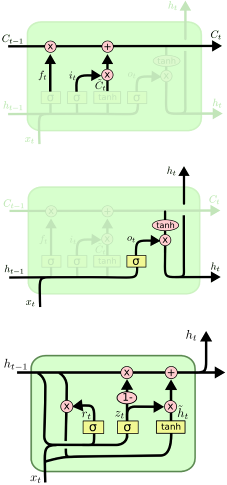


final memory cell: $c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t$

output gate: $o_t = \text{sigmoid}(\boldsymbol{A}_o[h_{t-1}, x_t] + b_o)$

$h_t = o_t \cdot \tanh(c_t)$





LASSO and networks

see Meinshausen & Bühlmann (2006, High-dimensional graphs and variable selection with the Lasso), or Friedman *et al.* (2008, Sparse inverse covariance estimation with the graphical lasso)

Which components of $\boldsymbol{\Sigma}^{-1}$ are not equal to 0 ?
Consider a sample $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ from $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$. Let $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$
Let $\boldsymbol{S}$ denote the empirical covariance matrix,

$$\boldsymbol{S} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^\top$$

As in Banerjee *et al.* (2008, Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data), maximize log-likelihood (Gaussian log-likelihood of the data, partially maximized with respect to the mean parameter)

$$\log\left[\det(\boldsymbol{\Theta})\right] - \text{trace}[\boldsymbol{S}\boldsymbol{\Theta}] - \lambda\|\boldsymbol{\Theta}\|_{\ell_1}$$

(for non-negative definite matrices $\boldsymbol{\Theta}$)

The objective function is

$$\underbrace{\log\left[\det(\boldsymbol{\Theta})\right] - \text{trace}[\boldsymbol{S\Theta}]}\; - \underbrace{\lambda\|\boldsymbol{\Theta}\|_{\ell_1}}_{\text{penalization}}$$

where $\|\boldsymbol{\Theta}\|_{\ell_1} = \sum \Theta_{i,j}$.

See van Wieringen (2016, Undirected network reconstruction from high-dimensional data)
and https://github.com/kaizhang/glasso for graphical lasso.
source: http://khughitt.github.io/