Report

Degrees of freedom (DF)

Degrees of freedom can used for obtaining an unbiased estimation of the error variance.

$$df(\hat{\mu}) = \frac{1}{\sigma_e^2}\sum_{i=1}^{N} Cov(z_i, \mu_i) = \sum_{i=1}^{N} E\left[\frac{\partial \hat{\mu}_i(z)}{\partial z_i}\right]$$

Consider the linear model:

$$Y = X\beta + \epsilon,\ \epsilon \sim N(0, \sigma^2 I)$$

Using the least squares method, it can be estimated by:

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$
$$\hat{\mu} = X(X^T X)^{-1} X^T Y$$

Degrees of freedom is:

$$df(\hat{\mu}) = \sum_{i=1}^{N} E\left[\frac{\partial \hat{\mu}_i(y)}{\partial y_i}\right] = trace\left(X(X^T X)^{-1} X^T\right) = p$$

p is the number of variables.

To measure how well the model captures the underlying structure, consider the quadratic loss:

$$(\hat{\mu} - \mu)^T(\hat{\mu} - \mu),\ \ u = X\beta$$

Akaike (1973) showed that the quadratic loss can be estimated unbiasedly by(AIC):

$$AIC = (\hat{\mu} - y)^T(\hat{\mu} - y) - n\sigma^2 + 2 * df * \sigma^2$$

Another useful criterion is the analog to GCV (Craven and Wahba 1979):

$$GCV = (\hat{\mu} - y)^T(\hat{\mu} - y)/(n - df)$$

Generalized degrees of freedom (GDF)

Degrees of freedom can be estimated by Monte Carlo method in the equation:

$$\begin{aligned} GDF(\hat{\mu}) &= \sum_{i=1}^{N} E\left[\frac{\partial \hat{\mu}_i(y)}{\partial y_i}\right] = \sum_{i=1}^{N} \lim_{\delta \to 0} E\left[\frac{\hat{\mu}_i(y + \delta e_i) - \hat{\mu}_i(y)}{\delta}\right] \\ &= \frac{1}{\sigma^2}\sum_{i=1}^{N} E\left[\hat{\mu}_i(y)(y_i - \mu_i)\right] \\ &= \frac{1}{\sigma^2}\sum_{i=1}^{N} cov(\hat{\mu}_i(y), y_i - \mu_i) \end{aligned}$$

To estimate GDF, use Monte Carlo method:

Algorithm 1

- Repeat $t = 1, \ldots, T$.
- Generate $\Delta_t = (\delta_{t1}, \ldots, \delta_{tn})$ from the density $\prod(1/\tau^n)\phi(\delta_{ti}/\tau)$.
- Evaluate $\hat{\mu}_i(Y + \Delta_t)$ based on the modeling procedure $\mathcal{M}$.
- Calculate $\hat{h}_i^{\mathcal{M}}$ as the regression slope from

$$\hat{\mu}_i(\mathbf{Y} + \Delta_t) = \alpha + \hat{h}_i \delta_{ti}, \qquad t = 1, \ldots, T,$$

So

$$GDF = \sum_i \hat{h_i}$$

The choice of parameter:

Let $\phi(\cdot)$ be an n-variate standard normal density function. We consider the estimate of $D(\mathcal{M})$ to be of the form

$$\hat{D}(\mathcal{M}) = \sum \frac{\partial E_{\mathbf{Y}}^{*}[\hat{\mu}_i(\mathbf{Y}^{*})]}{\partial y_i} = \frac{1}{\tau^2} \int \boldsymbol{\delta}' \hat{\boldsymbol{\mu}}(\mathbf{Y} + \boldsymbol{\delta}) \frac{1}{\tau} \phi\left(\frac{\boldsymbol{\delta}}{\tau}\right) d\boldsymbol{\delta}, \tag{A.1}$$

where the expectation $E_{\mathbf{Y}}^{*}$ is taken under the distribution $\mathbf{Y}^{*} \sim \mathrm{N}(\mathbf{Y}, \tau^2 \mathbf{I})$. Note that

$$\begin{aligned} E\hat{D}(\mathcal{M}) &= \sum \frac{\partial E_{\mathbf{Y}}^{**}[\hat{\mu}_i(\mathbf{Y}^{**})]}{\partial \mu_i} \\ &= \frac{1}{\sqrt{\sigma^2 + \tau^2}} \int \varepsilon' \hat{\boldsymbol{\mu}}(\boldsymbol{\mu} + \varepsilon) \frac{1}{\sqrt{\sigma^2 + \tau^2}} \phi\left(\frac{\varepsilon}{\sqrt{\sigma^2 + \tau^2}}\right) d\varepsilon, \end{aligned} \tag{A.2}$$

where the expectation E^{**} is taken with respect to the distribution $\mathbf{Y}^{**} \sim \mathrm{N}(\boldsymbol{\mu}, (\tau^2 + \sigma^2)\mathbf{I})$. From (A.2), we have

Proposition 2. $\lim_{\tau \to 0} E_{\boldsymbol{\mu}} \hat{D}(\mathcal{M}) = D(\mathcal{M})$.

Note that Proposition 2 is practically irrelevant. Consider the decomposition

$$E[\hat{D}(\mathcal{M}) - D(\mathcal{M})] = \mathrm{var}(\hat{D}(\mathcal{M})) + [E\hat{D}(\mathcal{M}) - D(\mathcal{M})]^2. \tag{A.3}$$

As $\tau \to 0$, $\mathrm{var}(\hat{D}(\mathcal{M}))$ goes to ∞ when $\hat{\boldsymbol{\mu}}(\mathbf{Y})$ is highly nonlinear locally, as in Example 1, or discontinuous. Thus it is often desirable to avoid $\tau \sim 0$. On the other hand, if $\tau \gg \sigma$, then the bias $[E\hat{D}(\mathcal{M}) - D(\mathcal{M})]^2$ may increase. Empirically, the GDF is not sensitive to τ for $\tau \in [.5\sigma, \sigma]$.

Extended AIC(EAIC):

$$EAIC = (\hat{\mu} - y)^T (\hat{\mu} - y) - n\sigma^2 + 2 * GDF * \sigma^2$$

Extended GCV:

$$GCV = (\hat{\mu} - y)^T (\hat{\mu} - y)/(n - GDF)$$

Analyze GDF for linear model

```
import numpy as np
from scipy import stats

def D(T,X,Y,a,func):
    n = len(Y)
    delta_t = np.zeros(n)
    delta = np.zeros((T,n))
    u = np.zeros((T,n))
    for t in range(T):
        tmp = np.random.randn(n)*a
        delta_t = stats.norm.pdf(tmp/a,0,1)/a
        beta = func(X,Y+delta_t)
        u[t,:] = (X@beta).T
        delta[t,:] = delta_t
    ans = 0
```

```
    mean = np.mean(delta,axis=0)
    mean_u = np.mean(u,axis=0)
    for i in range(n):
        hi = np.dot(delta[:,i]-mean[i],u[:,i]-mean_u[i])/np.dot(delta[:,i]-
mean[i],delta[:,i]-mean[i])
        ans += hi
    return ans

def func(X,Y):
    return np.linalg.inv(X.T@X)@X.T@Y
```

First I analyze linear model （see code in "Linear model.py"）, I set

$$Y = X\beta + \epsilon$$
$$p = 5$$
$$n = 100$$
$$\sigma_e = 0.5$$

The estimate GDF is 5.008294096432965, which is similar to 5(true answer).

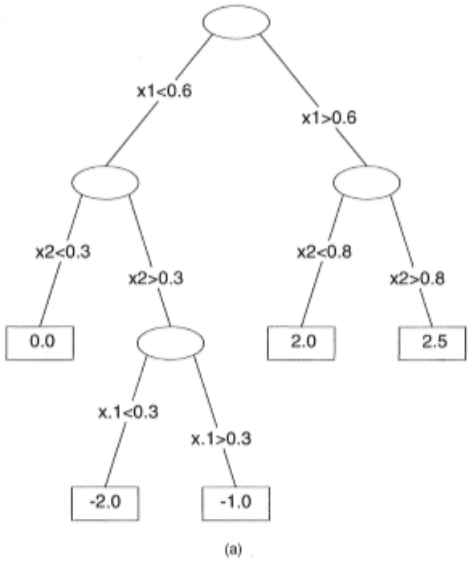
Reproduct Nonparametric Regression

Consider the following models:

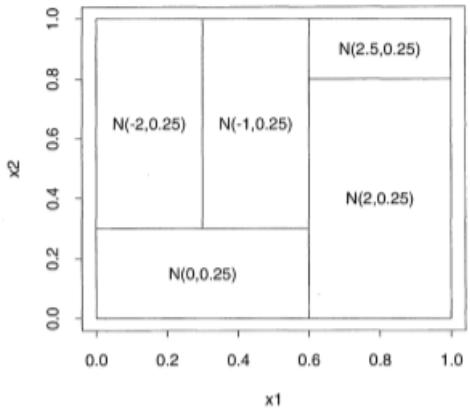
$$N_2\colon\ y = 0 \cdot x_1 + 0 \cdot x_2 + \varepsilon,$$

$$N_{10}\colon\ y = 0 \cdot (x_1 + \cdots + x_{10}) + \varepsilon,$$

$$T_{10}\colon\ y = s(x_1, x_2) + 0 \cdot (x_3 + \cdots + x_{10}) + \varepsilon.$$



(a)



Using classification and regression trees(CART), degrees of freedom equal to the number of terminal node Nt, so variance estimate can be

$$\hat{\sigma_o^2} = \frac{(Y-\hat{\mu})^T(Y-\hat{\mu})}{n-N_t}$$

It can be shown that if a CART overfits the data, then the variance estimate when GDF is used,

$$\hat{\sigma^2} = \frac{(Y-\hat{\mu})^T(Y-\hat{\mu})}{n-g}$$

gives an unbiased estimate of the error variance.

Table 1——see code in "Applications to nonparameter regression_table1.py"

Models	Nodes	GDF	sigma_o^2/s^2	sigma^2/s^2
N2	19	61.76(0.86)	0.47(0.00)	1.00(0.02)
N10	19	79.42(0.44)	0.25(0.00)	1.00(0.02)
T10	19	76.52(0.68)	0.32(0.00)	1.11(0.02)

The parameter setting is the same in the paper:

Table 1. Variance Estimate in CART With Fixed Number of Terminal Nodes

Model	*Nodes*	*GDF*	$\hat{\sigma}_o^2/s^2$	$\hat{\sigma}^2/s^2$
N_2	19	65.2(.78)	.43(.05)	1.01(.15)
N_{10}	19	79.1(.44)	.25(.04)	.97(.19)
T_{10}	19	75.9(1.14)	.31(.04)	1.04(.20)

NOTE: $n = 100$, $\sigma = .5$, $\tau = .5\sigma$; results are based on 100 simulations.

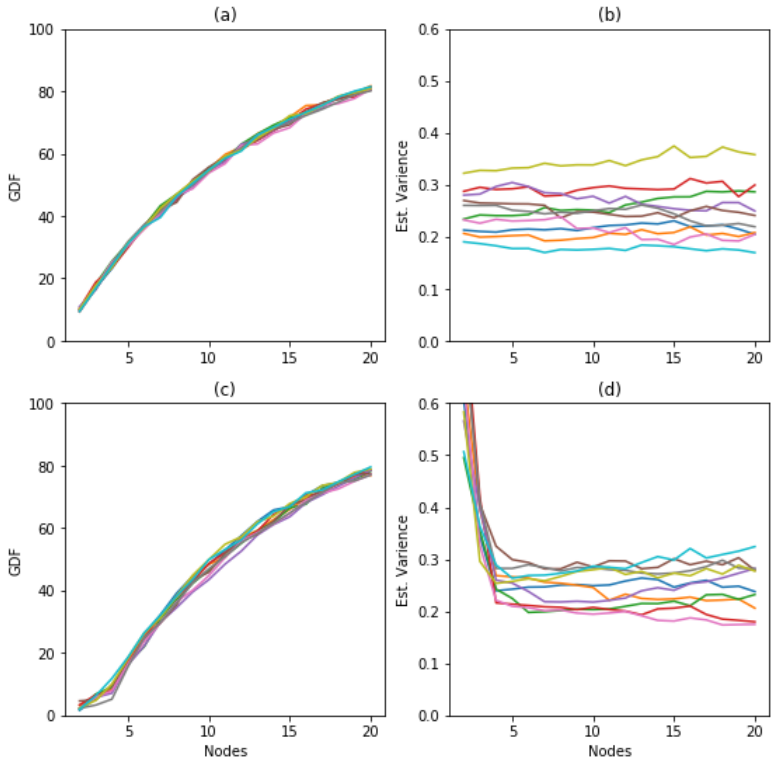
Where

$$s^2 = (1/N)\sum_i \epsilon_i^2$$

Table 1 shows the estimate \sigma_o^2 biases downward substantially. The new variance estimate, \sigma^2, is almost unbiased for all three cases in Table 1.

The GDF are larger for model N10 than for model N2 due to more independent variables, even though the sizes are the same. With 100 observations and 10 variables, a tree of 19 nodes uses up to 79 df.

Figure 2——see code in "Applications to nonparameter regression_figure2.py"



The parameter setting is the same in the paper:

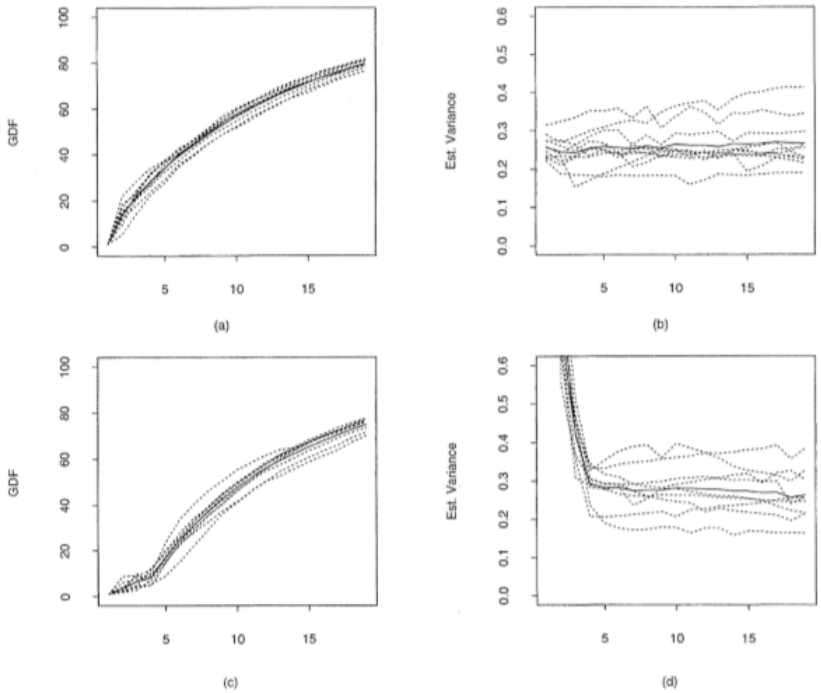


Figure 2. The GDF and $\hat{\sigma}^2$ for the Estimated Trees of Different Sizes. (a) and (b) Ten simulations based on model N_{10}; (c) and (d) ten simulations based on T_{10}. Results are obtained wtih $n = 100$, $\sigma = 0.5$, and $\tau = .5\sigma$.

Figure 2 shows the behavior of GDF and \sigma^2 for fitted trees of different size. The estimate \sigma^2 stabilizes near 0.25 when the tree is large enough to fit the true models.

Table 2——see code in "Applications to nonparameter regression_table2.py"

Model	1	2	3	4	5
N10	9.98	17.49	24.41	30.54	36.20
T10	2.55	6.04	9.50	18.19	25.40

The parameter setting is the same in the paper:

Table 2. GDF for Two Different Models

Model	*1*	*2*	*3*	*4*	*5*
N_{10}	14.8	22.6	29.2	34.7	39.8
T_{10}	2.7	5.6	8.18	17.6	25.4

NOTE: The settings are the same as in Table 1.

Table 2 displays GDF for models N10 and T10. Fitting a tree to a pure noise dataset costs substantially more GDF than fitting to a dataset with a clear structure for the first several nodes. Searching for one node in a 10-dimensional space costs as many as 10 GDF when the data are pure noise.

Disscussion

If the fitted tree is large enough, the GDF is stable across different simulations (its variance is negligible). If the tree is too small to fit the underlying structure, then the variance of GDF can be considerably larger.

The sightly different results between paper and reproduction(In table 2 experiment) can caused by different packege.

Author used S-PLUS software tree( ), as described by Clark and Pregibon (1992) and Ven ables and Ripley in R Language. I use python language use package sklearn.tree.DecisionTreeRegressor( ), this can caused slightly difference in default parameter setting.

Reproduct Selected Model

Consider a linear model that has q variables:

$$Y = \sum_{j=1}^{q} \beta_j X_j + \epsilon$$

Let

$$M = \{x_{j,1}, \ldots, x_{j,k}\}$$
$$\hat{Y_M} = \sum_{h=1}^{k} \beta_{jh} x_{jh}$$

For a fixed 0 < k < q, select Mk as

$$M_k = \arg\min_M (\hat{Y_M} - Y)^T (\hat{Y_M} - Y)$$

To evaluate the goodness of Mk, define a modeling procedure Mk as a combination of model selection.

$$\mathcal{M}_k: \mathbf{Y} \xrightarrow{\text{selection}} M_k(\mathbf{Y}) \xrightarrow{\text{fitting } M_k(\mathbf{Y})} \hat{\mathbf{Y}}_{M_k}.$$

Let D(k) = D(Mk). Then GDF can be applied to Mk to obtain an unbiased assessment of the model Mk.

Example 4. Data are generated from a linear model

$$y_i = \alpha + \sum_{j=1}^{q} \beta_i x_{ij} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

where $q = 20$, $n = q + 2 = 22$, and ε_i's are iid as N(0, 1); $\alpha = 0$ is set in all cases. The coefficient vector $\beta = (\beta_1, \ldots, \beta_{20})$ is set to take one of the following two values:

$$\beta^{(0)} = 0_{20} : \beta_1 = \cdots = \beta_{20} = 0;$$

or

$$\beta^{(1)} = 2_5 0_{15} : \beta_1 = \cdots = \beta_5 = 2, \quad \beta_6 = \cdots = \beta_{20} = 0.$$

Let $\mathbf{x}_{i\cdot} = (x_{i1}, \ldots, x_{ip})$ be the ith row of the matrix $\mathbf{X} = (x_{ij})_{22\times 20}$. $\mathbf{X}$ is generated from

$$\mathbf{x}_{i\cdot} \overset{\text{iid}}{\sim} \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}), \qquad i = 1, \ldots, n,$$

where $\boldsymbol{\Sigma} = (1 - \rho)\mathbf{I} + \rho\mathbf{J}$, $\mathbf{I}$ is an identity matrix and $\mathbf{J}$ is a matrix of 1s.

All cases use the perturbation size $\tau = .5\sigma$, with the number of perturbations $T = 100$. The results are not sensitive to τ for $\tau \in [.5\sigma, \sigma]$. The intercept is always included in selected models.

Table 3 case 1——see code in "Selected Model_table3_case1.py"

K	1	5	6	10	15	20
GDF	3.72	13.38	13.71	16.85	18.67	19.83
AIC	-0.94	-2.72	-1.58	3.76	12.83	22.37
Loss	2.28	12.06	12.92	15.58	16.50	16.97
EAIC	2.49	12.03	11.84	15.47	18.16	20.03
s^2(adj)	0.85	0.45	0.43	0.34	0.47	2.37
s^2(cor)	0.93	0.84	0.77	0.73	0.85	1.09
R^2(adj)	0.03	0.48	0.51	0.61	0.46	-1.70
R^2(cor)	-0.06	0.04	0.12	0.17	0.03	-0.24

Table 3 case 2——see code in "Selected Model_table3_case2.py"

K	1	5	6	10	15	20
GDF	20.34	5.27	7.62	16.52	19.69	20.05
AIC	400.72	7.83	5.15	5.26	11.69	21.20
Loss	440.40	4.47	9.15	17.05	20.61	21.10
EAIC	437.39	6.37	6.39	16.30	19.08	19.31
s^2(adj)	20.94	1.11	0.88	0.48	0.28	1.20
s^2(cor)	251.71	1.07	0.91	0.96	0.73	0.62
R^2(adj)	0.78	0.99	0.99	1.00	1.00	0.99
R^2(cor)	-1.61	0.99	0.99	0.99	0.99	0.99

The parameter setting is the same in the paper:

Table 3. Goodness-of-Fit Statistics of a Selected Model: Bias and the Corresponding Corrections

	k (Dimensionality of the model)					
	1	*5*	*6*	*10*	*15*	*20*
			Case 1 ($\beta = 0_{20}$)			
GDF	5.23	14.10	15.37	18.75	20.56	21.00
AIC	−1.28	−2.21	−1.48	3.24	11.37	20.96
Loss	5.24	14.17	15.44	18.72	20.59	21.00
EAIC	5.18	13.99	15.26	18.74	20.48	20.95
s^2(adj)	.84	.49	.44	.29	.23	.96
s^2(cor)	1.00	1.02	1.03	1.10	1.00	.91
R^2(adj)	.16	.51	.56	.71	.77	.05
R^2(cor)	0	−.02	−.03	−.09	.01	.09
			Case 2 ($\beta = 2_5 0_{15}$)			
GDF	10.98	6.32	10.25	17.43	20.51	21.00
AIC	347	6.22	4.20	4.79	11.59	20.99
Loss	362	6.27	10.33	17.76	20.84	21.33
EAIC	365	6.86	10.70	17.66	20.61	21.00
s^2(adj)	18.23	1.01	.81	.44	.26	.99
s^2(cor)	88.96	1.05	1.06	1.18	1.14	.94
R^2(adj)	.69	.98	.99	.99	1.00	.98
R^2(cor)	−.53	.98	.98	.98	.98	.98

NOTE: $n = 22$, $q = 20$, $\rho = .5$, $\sigma = 1$, and $\tau = .5$. Results are based on 1,000 simulations with the same design.

Table 3 illustrates the bias from ignoring the selection process and performance of the GDF, where:

$$
\begin{aligned}
AIC &= (\hat{\mu}-Y)^T(\hat{\mu}-Y) - n\sigma^2 + 2*df*\sigma^2 \\
Loss &= (\hat{\mu}-\mu)^T(\hat{\mu}-\mu),\ \ \mu = X\beta \\
EAIC &= (\hat{\mu}-Y)^T(\hat{\mu}-Y) - n\sigma^2 + 2*GDF*\sigma^2 \\
s^2(adj) &= \frac{(\hat{\mu}-Y)^T(\hat{\mu}-Y)}{n-df} \\
s^2(cor) &= \frac{(\hat{\mu}-Y)^T(\hat{\mu}-Y)}{n-GDF} \\
R^2(adj) &= 1 - \frac{s^2(adj)}{Y^TY/n} \\
R^2(cor) &= 1 - \frac{s^2(cor)}{Y^TY/n}
\end{aligned}
$$

Table 3 shows that s^2(adj) becomes substantially smaller than \sigma=1 as k increases. R^2(adj) increases as k increases. In case 1 the average R^2(adj) reaches 46% even though \beta= 0. The estimates using GDF behave substantially better. The s^2(cor) is nearly unbiased for \siama when k>=0 in case 1 and k>=5 in case 2. In case 1, R^2(cor) is near 0 for all k. In case 2, R^2(cor) stablizes when k>=5, because the true model has five variables with nonzero coefficients.

Table 4 case 1.1——see code in "Selected Model_table4_case1.1.py"

beta	0_20
	True Loss
EAIC	2.59
AIC	8.60
GCV_gdf	4.51
GCV^*_gdf	3.74
GCV_df	14.08
GCV^*_df	13.84
	Average number of variables selected
EAIC	0.56
AIC	2.31
GCV_gdf	1.78
GCV^*_gdf	1.11
GCV_df	7.49
GCV^*_df	7.07
	s^2(adj)
GCV_df	0.46
GCV^*_df	0.47
	s^2(cor)
GCV_gdf	0.81
GCV^*_gdf	0.84
GCV_df	0.71
GCV^*_df	0.73

Table 4 case 1.2——see code in "Selected Model_table4_case1.2.py"

beta	2_5 0_15
	True Loss
EAIC	6.96
AIC	11.66
GCV_gdf	8.78
GCV^*_gdf	7.82
GCV_df	15.12
GCV^*_df	14.83
	Average number of variables selected
EAIC	5.56
AIC	7.21
GCV_gdf	6.96
GCV^*_gdf	6.13
GCV_df	10.37
GCV^*_df	9.97
	s^2(adj)
GCV_df	0.55
GCV^*_df	0.58
	s^2(cor)
GCV_gdf	0.83
GCV^*_gdf	0.89
GCV_df	0.69
GCV^*_df	0.75

Table 4 case 2.1——see code in "Selected Model_table4_case2.1.py"

beta	0_20
	True Loss
EAIC	2.91
AIC	9.36
GCV_gdf	4.90
GCV^*_gdf	3.86
GCV_df	14.21
GCV^*_df	14.11
Average number of variables selected	
EAIC	0.54
AIC	2.56
GCV_gdf	2.18
GCV^*_gdf	1.02
GCV_df	7.35
GCV^*_df	7.06
	s^2(adj)
GCV_df	0.45
GCV^*_df	0.46
	s^2(cor)
GCV_gdf	0.79
GCV^*_gdf	0.84
GCV_df	0.72
GCV^*_df	0.75

Table 4 case 2.2——see code in "Selected Model_table4_case2.2.py"

beta	2_5 0_15
	True Loss
EAIC	8.25
AIC	12.77
GCV_gdf	10.11
GCV^*_gdf	8.56
GCV_df	16.11
GCV^*_df	15.20
Average number of variables selected	
EAIC	5.78
AIC	7.58
GCV_gdf	7.75
GCV^*_gdf	6.16
GCV_df	11.06
GCV^*_df	9.96
	s^2(adj)
GCV_df	0.55
GCV^*_df	0.62
	s^2(cor)
GCV_gdf	0.81
GCV^*_gdf	0.90
GCV_df	0.65
GCV^*_df	0.73

Table 4

beta	0_20
	True Loss
EAIC	2.59
AIC	8.60
GCV_gdf	4.51
GCV^*_gdf	3.74
GCV_df	14.08
GCV^*_df	13.84
Average number of variables selected	
EAIC	0.56
AIC	2.31
GCV_gdf	1.78
GCV^*_gdf	1.11
GCV_df	7.49
GCV^*_df	7.07
	s^2(adj)
GCV_df	0.46
GCV^*_df	0.47
	s^2(cor)
GCV_gdf	0.81
GCV^*_gdf	0.84
GCV_df	0.71
GCV^*_df	0.73

beta	2_5 0_15
	True Loss
EAIC	6.96
AIC	11.66
GCV_gdf	8.78
GCV^*_gdf	7.82
GCV_df	15.12
GCV^*_df	14.83
Average number of variables selected	
EAIC	5.56
AIC	7.21
GCV_gdf	6.96
GCV^*_gdf	6.13
GCV_df	10.37
GCV^*_df	9.97
	s^2(adj)
GCV_df	0.55
GCV^*_df	0.58
	s^2(cor)
GCV_gdf	0.83
GCV^*_gdf	0.89
GCV_df	0.69
GCV^*_df	0.75

beta	0_20
	True Loss
EAIC	2.91
AIC	9.36
GCV_gdf	4.90
GCV^*_gdf	3.86
GCV_df	14.21
GCV^*_df	14.11
Average number of variables selected	
EAIC	0.54
AIC	2.56
GCV_gdf	2.18
GCV^*_gdf	1.02
GCV_df	7.35
GCV^*_df	7.06
	s^2(adj)
GCV_df	0.45
GCV^*_df	0.46
	s^2(cor)
GCV_gdf	0.79
GCV^*_gdf	0.84
GCV_df	0.72
GCV^*_df	0.75

beta	2_5 0_15
	True Loss
EAIC	8.25
AIC	12.77
GCV_gdf	10.11
GCV^*_gdf	8.56
GCV_df	16.11
GCV^*_df	15.20
Average number of variables selected	
EAIC	5.78
AIC	7.58
GCV_gdf	7.75
GCV^*_gdf	6.16
GCV_df	11.06
GCV^*_df	9.96
	s^2(adj)
GCV_df	0.55
GCV^*_df	0.62
	s^2(cor)
GCV_gdf	0.81
GCV^*_gdf	0.90
GCV_df	0.65
GCV^*_df	0.73

The parameter setting is the same in the paper:

Table 4. Dimension Selection: Comparisons of Criteria

	$\rho = .5$		$\rho = 0$	
β	0_{20}	$2_5 0_{15}$	0_{20}	$2_5 0_{15}$
		True loss		
EAIC	3.39	8.74	3.17	8.26
AIC	11.67	14.75	11.86	14.50
GCV_{gdf}	6.82	11.59	6.72	11.24
GCV^*_{gdf}	4.17	9.26	4.05	8.74
GCV_{df}	18.92	19.65	18.95	19.53
GCV^*_{df}	18.49	19.06	18.53	18.92
		Average number of variables selected		
EAIC	.40	5.50	.36	5.40
AIC	2.98	7.39	3.09	7.36
GCV_{gdf}	3.82	8.40	3.79	8.38
GCV^*_{gdf}	.84	5.76	.84	5.70
GCV_{df}	11.53	13.34	11.49	13.27
GCV^*_{df}	10.09	11.57	10.09	11.60
		s^2 *(adj)*		
GCV_{df}	.22	.25	.22	.26
GCV^*_{df}	.26	.31	.25	.31
		s^2 *(cor)*		
GCV_{gdf}	.75	.72	.75	.72
GCV^*_{gdf}	.88	.88	.88	.88
GCV_{df}	.54	.59	.54	.57
GCV^*_{df}	.65	.78	.65	.74

NOTE: $n = 22$, $p = 20$, $\sigma = 1$, and $\tau = .5$. Results are from 1,000 simulations with the same design.

while exploring the whole subset, it takes

$$C_{20}^0 + C_{20}^1 + \ldots + C_{20}^{20} = 20!$$

times calculation, which is unbearable in my computer. So I use Monte Carol Method, let:

$$Case\,1$$
$$\begin{cases} Explore\ whole\ M_k & else \\ Random\ 1000\ times & 4 <= K <= p-4 \end{cases}$$

$$Case\,2$$
$$\begin{cases} Explore\ whole\ M_k & else \\ Subset\ (x_1 \sim x_K) + Random\ 1000\ times & 4 <= K <= p-4 \end{cases}$$

Table 4 illustrates the performance of EAIC, AIC, GCV_gdf, and GCF_df as variable selection procedures. The table gives the true loss of the models selected by each criterion, the average number of variables in the selected model, and the variance estimates when degrees of freedom and GDF are used.

For the null model case 1, EAIC selects about 0.56 variable on average, whereas the model selected by AIC has 2.31 variables. In all cases simulated, the dimensionality selected by minimizing EAIC has a smaller average true loss than that selected by minimizing AIC. The improvement can often be substantial, especially in cases where the true model contains relatively few strong variables.

Disscussion

In this situation, n = p+2 = 22, the observation data size is too small. Since in every different situations, GDF can fluctuate a little bit. So it may be had to reproduct the result in paper. However, I can show the same relation and the same results, that is the performance:

$$EAIC > GCV_{gdf^*} > GDV_{gdf} > AIC > GCV_{df^*} > GDV_{df}$$

Conclude

Generalized degrees of freedom (GDF) is applicable for evaluation of the final model or fits produced by such a process. The GDF is defined as the sum of the sensitivity of each fitted value to perturbations in the corresponding observed value. It is nonasymptotic in nature and thus is free of the sample-size constrain.

There are several differences between GDF and the traditional degrees of freedom:

1. There is no longer an exact correspondence between the degrees of freedom and the number of parameters.
2. GDF depends on both the modeling procedure and the underlying true mode.

Simulations results shows GDF can obtain an unbiased estimate of variance，which help to choose the fittest model.