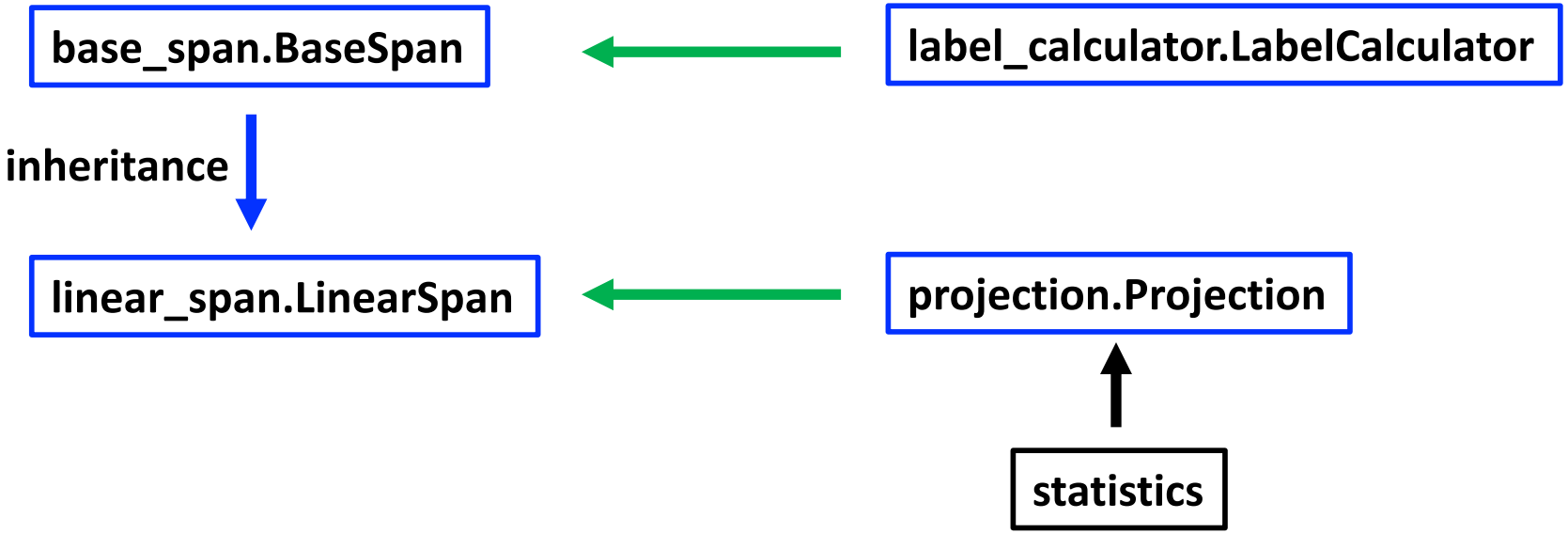
A basic manual of LIDG program

Hitoshi Fujii

Outline

- Configuration of LIDG program
- Workflow of interpretable sparse modeling by LIDG
- Flowchart
- Term description
- Methods
- The meaning of subspace list
- Definition of criteria
- An example of backward selection method

Configuration of LIDG program (1/2)



Classes:

- **BaseSpan: super class to handle pandas DataFrame.**
- **LinearSpan: main class (linearly independentize, generation, symmetrization, projection)**
- **LabelCalculator: label handling, calculation of strings.**
- **Projection: OLS, find near-multicollinearities.**

Configuration of LIDG program (2/2)

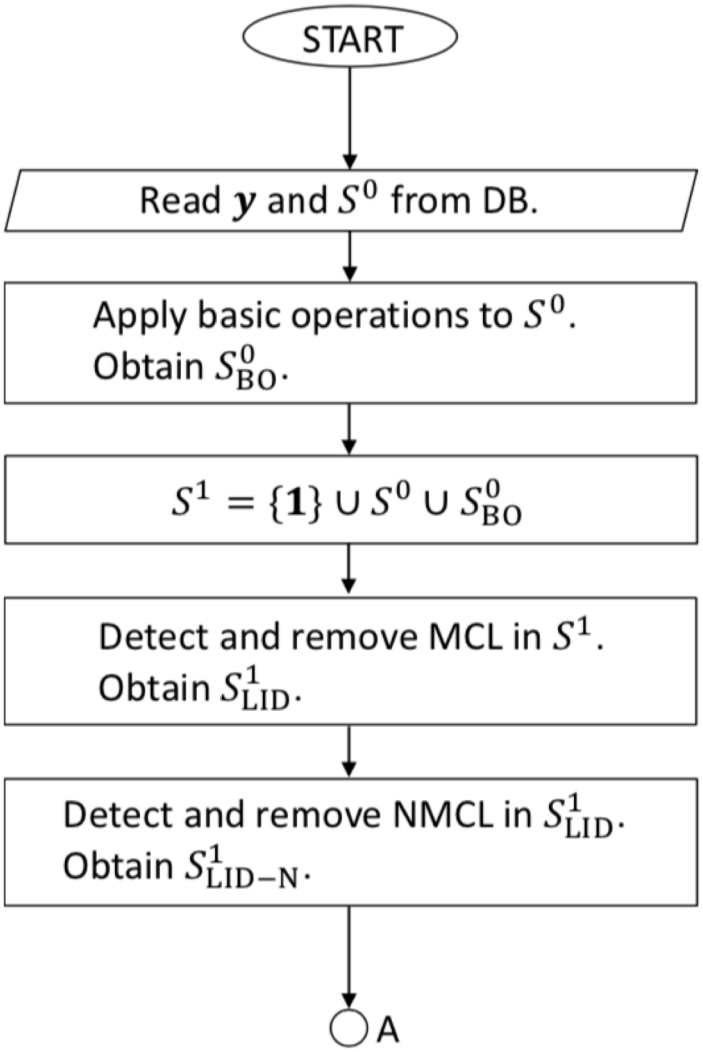
lidg/__init__.py

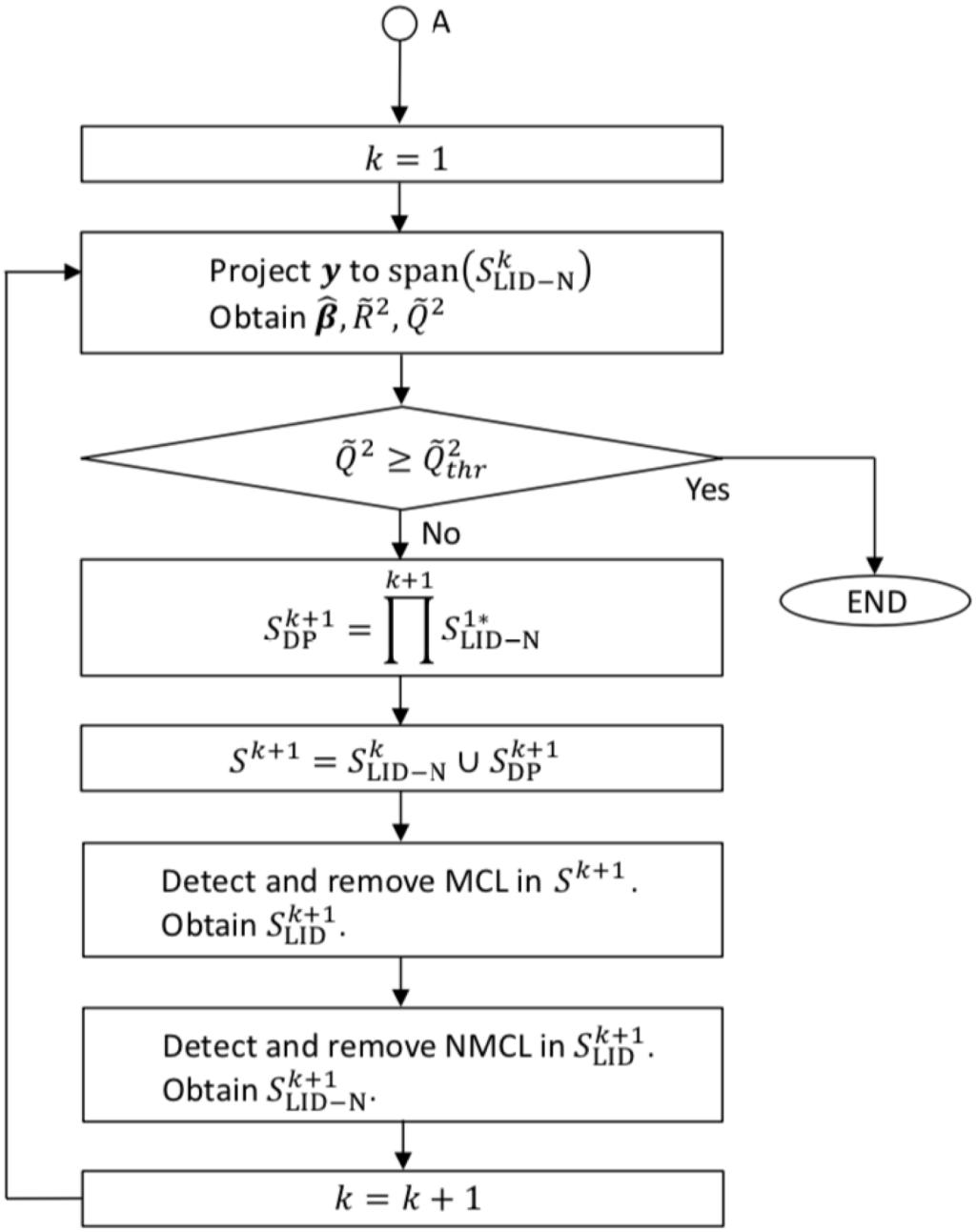
base_span.py	Super class of LinearSpan
label_calculator.py	Label handling, calculation of strings
linear_span.py	Main program
projection.py	Linear regression
multicollinearity.py	Find multicollinearities in span
statistics.py	Statistic calculations
skl_enet.py	Elastic net calculation by sklearn
plots.py	Plotting method
database.py	Handle a relational database

Workflow of interpretable sparse modeling by LIDG

1. Read target and descriptor vectors
2. Generate first order descriptors by basic operations (BO)
3. Find and remove multicollinearities (MCL)
4. Find and remove strong near multicollinearities (NMCL)
5. Perform OLS and check the accuracy of the model
6. Generate higher order descriptors by direct product (DP)
7. Go to 3. Repeat until the regression model reaches a sufficient accuracy
8. (perform a descriptor selection, such as BS, LASSO, MCTS, etc.)

Flowchart


START
Read $\boldsymbol{y}$ and S^0 from DB.
Apply basic operations to S^0.
Obtain S^0_{BO}.
$S^1 = \{\mathbf{1}\} \cup S^0 \cup S^0_{\mathrm{BO}}$
Detect and remove MCL in S^1.
Obtain S^1_{LID}.
Detect and remove NMCL in S^1_{LID}.
Obtain $S^1_{\mathrm{LID-N}}$.
A


A
$k = 1$
Project $\boldsymbol{y}$ to span$(S^k_{\mathrm{LID-N}})$
Obtain $\widehat{\boldsymbol{\beta}}, \tilde{R}^2, \tilde{Q}^2$
$\tilde{Q}^2 \geq \tilde{Q}^2_{thr}$
Yes
No
END
$S^{k+1}_{\mathrm{DP}} = \prod^{k+1} S^{1*}_{\mathrm{LID-N}}$
$S^{k+1} = S^k_{\mathrm{LID-N}} \cup S^{k+1}_{\mathrm{DP}}$
Detect and remove MCL in S^{k+1}.
Obtain S^{k+1}_{LID}.
Detect and remove NMCL in S^{k+1}_{LID}.
Obtain $S^{k+1}_{\mathrm{LID-N}}$.
$k = k + 1$

Term description

線形包（Linear span）または単にスパン:
数学用語。ベクトルの集合であり、それらが張るベクトル空間のこと。

サンプル空間、目的変数ベクトル、記述子ベクトル、記述子行列、記述子空間:
$\boldsymbol{m}$個のサンプル物質（の様々な物性値に対する値）を直交基底とした$\boldsymbol{m}$次元サンプル空間$\boldsymbol{V}$を考える。ある一つの物性値を目的変数とすると、その各サンプル物質に対する値の配列はサンプル空間上の$\boldsymbol{m}$次元ベクトルとみなせる。これを目的変数ベクトル$\boldsymbol{y}$と呼ぶことにする。同様に、他の$\boldsymbol{n}$個の物性値（記述子）の各サンプルに対する値もサンプル空間上のベクトルと見なす。これら$\boldsymbol{n}$個の$\boldsymbol{m}$次元ベクトルを記述子ベクトル$\boldsymbol{x}$と呼ぶことにする。これら$\boldsymbol{n}$個のベクトルを列ベクトルとした$\boldsymbol{m}\times\boldsymbol{n}$行列を記述子行列（計画行列）$\boldsymbol{X} \equiv [\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}]$と呼ぶことにする。記述子ベクトルが張る空間、つまりスパン$\boldsymbol{S} \subset \boldsymbol{V}$を記述子空間と呼ぶことにする。
$\mathbf{rank}(\boldsymbol{X}) = \mathbf{dim}(\boldsymbol{S})$ が成り立つ。

最小二乗法（OLS）:
幾何学的には、スパン$\boldsymbol{S}$（超平面）に目的変数ベクトル$\boldsymbol{y}$を射影（vector projection）したベクトルを見つけることに他ならない。（$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$）

残差（residual）:
vector rejection、つまり、OLSでの推定値$\hat{\boldsymbol{y}}$から目的変数ベクトル$\boldsymbol{y}$へのベクトル（$\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}}$）。

Methods (1/6) (basic methods)

.read()
スペース区切りテキストもしくはpandasデータフレームを読み込む。

.set_y()
記述子と目的変数に分けられ、それぞれ内部変数（クラス変数）self.x、self.yに格納される。

.add_const()
定数項ベクトルが付け加えられる。

.change_label(str)
記述子のラベルを通し番号付きの指定したラベルに変更。

.show()
スパン内の記述子ベクトルの数、それらのラベルを表示する。

.show_save()
バックアップ用スパン内の記述子ベクトルの数、それらのラベルを表示する。

スパンに関するクラス変数は3つあり、一つはバックアップ用のもの（self.x_save）で、もう一つは作業用のもの（self.x）である。実際の使用では、作業用スパン内の記述子を削除したり新たな記述子を追加して回帰精度などを逐一確認するというようなことが頻繁に行われるので、このように、いつでも変更前に戻れる様、バックアップのためのクラス変数を用意している。最後の3つ目のクラス変数はself.originであり、これは新たな記述子を作る際に使われる。.readで読み込まれた時点ではこれら3つは全て同じである。また、スパンをいくら変更しても、self.originが変更されることはない。（.new()メソッドを除いて。）

Methods (2/6) (basic methods)

.save()
作業用スパンの記述子（self.x）をバックアップ用スパン（self.x_save）へコピーする。

.load(str or list)
バックアップ用スパン（self.x_save）内の指定した記述子を作業用スパン（self.x）へ追加する。

.remove(str or list)
指定した記述子を作業用スパンから削除する。バックアップはそのまま。
オプション “const”: 定数ベクトルを見つけて削除する。
“inf”: 発散した要素を持つ記述子ベクトルを見つけて削除する。
“all”: 全ての記述子を削除する。

.add(str or list)
文字列（リスト）を与えると、self.origin内の記述子に基づいて記述子を新たに作り、追加する。内部で一旦ラベルを変更する前のオリジナルのラベルへ戻すので、ラベルの変更を気にせず記述可能。（self.originにある記述子以外の演算は不可能。ただし、実数は使える。）

.add_rh()
一様乱数ベクトルをスパンに追加。色々なテストに使える。

.add_rg()
ガウシアンベクトルをスパンに追加。色々なテストに使える。

.plot(df,series), **.plot_vs(series1,series2)**: pandas.DataFrame、pandas.Seriesのプロット

Methods (3/6) (generation, symmetrization)

.new(renew=True): return spn
元のスパンから新たなスパンを作る。renew=Trueとすると、新たなスパンのself.xがself.originへコピーされる。つまり、ラベルを変更していた場合、それがoriginへ反映されてしまい、もはや最初のラベルの情報が失われる。しかし、これにより、一番最初のラベルにいちいち戻って記述子を計算しなくてよくなるので、計算時間を短縮できる様になる。
.sub(str or list): return spn
スパンから部分スパンを返す。
.join(str or list): return spn
スパンを追加したスパンを返す。

.gen_bo(str or list): return spn
基本演算により記述子を生成する。
“r”は逆数、“a”は絶対値、”b”は絶対値の逆数。
”r+”（”r-”）は記述子の和（差）の逆数。全ての組み合わせを計算。
”a+”,”b+”も同様。
label_calculator.pyの中で、新たな基本演算を自分で定義できる。

.gen_dp(int): return spn
直積により記述子を生成する。（高次記述子生成）

.sym(list): return spn
対称化した記述子を返す。これを使うには、記述子のラベルはx(0,1,2,..)の様に識別子が明記されてないといけない。引数には[(0,1,2,..),(1,0,2,...),(0,2,1,..),...]の様に置換の形で恒等操作を含めた全ての対称操作を与える。これらが群を成してないとエラーとなる。

Methods (4/6) (multicollinearity)

.lid(normalize=False,tole1,tole2): return spn
線型独立化したスパンを返す。normalize=Trueとすると多重共線性関係の出力が規格化した記述子に対するものになる。（内部では数値安定性のため、全ての記述子は一旦規格化される）
tole1が小さいと、なるべく初期の記述子の順番を守りながら多重共線性関係を見つける様になる。もし、左から順に単純な形の記述子を並べている場合（推奨）、その単純な形の記述子を優先的に残すことができる。多重共線性関係にある記述子は、統計的にはどれが良い記述子か判断できないのであるが、解釈可能なモデリングという観点から、単純な形の記述子は複雑なものより価値が高いと判断する。
tole2を大きくすると検出する線形独立（と見なされる）記述子の数が減る。つまり、余分に記述子が削られることになり、結果として、強い準多重共線性をも削除することができる。よほど自明な多重共線性関係でない限り、多重共線性と準多重共線性との違いは（数値的には）曖昧なのである。

多重共線性がある場合の記述子選択の指標としては「単純さ」の他に、「超体積の大きさ」、「yとの相関の強さ」などが考えられる。

.volume_comb(defi_list,cand_list,k,ntop=None)
defi_listで必ず使用する記述子を指定し、cand_listの中からk個の候補を選び、それらの記述子を辺とする超体積を計算し、上位ntopの組み合わせを出力する。

.xty()
各記述子とyとの相関$X^T y$を計算する。

Methods (5/6) (near multicollinearity)

.project(sort=None, normalize=False)

OLSを行う。normalize=Trueとすると、規格化した記述子ベクトルに対する回帰係数を出力する。これにより、回帰係数の大きさをフェアに比較することができる様になる。推定値がself.yに格納される。

推定精度の指標

e2: 残差平方和、mse: mean square error（e2をサンプル数で割ったもの）、rmse: mseの平方根

R2: 決定係数

予測精度の指標

eq2: 予測残差、mseq: 予測残差をサンプル数で割ったもの、rmseq: mseqの平方根

Q2: 予測に対する決定係数（leave-one-out cross validationの結果と同じになる。）

AIC: 赤池情報量基準

TR2, TQ2：それぞれ、自然なR2、自然なQ2。これらは、yの分散（にmをかけたもの）の代わりに、yのノルム二乗でe2、qe2を割ったもの。オリジナルの指標であり、R2、Q2と似た振る舞いをする。

b, |b|: 回帰係数とその絶対値。

p_val, -log10(p) : 回帰係数のT検定のp値と、それの対数を取ったもの。, -log10(p)が大きいほど重要な記述子と判断できる。

G2: 記述子の重要度の指標。規格化した記述子に対する回帰係数の大きさ、-log10(p)と同じ振る舞いをすると期待される。

TRi2: 記述子xiを残りの記述子を使って回帰した際の決定係数。準多重共線性の強さを表す。

Methods (6/6)

.rij(ntop)
ピアソンの対相関係数（標準化した記述子同士の内積）の大きいペアからntopまで出力。

.nij(ntop)
規格化した記述子同士の内積を計算し、大きいものからntopまで出力。

.gij(ntop)
一般化対相関係数（オリジナルの指標）。ピアソンの対相関係数では準共線性しか検出できないが、この一般化対相関係数は準多重共線性も考慮した指標となっている。

.gik(str,ntop)
ある一つの記述子を指定し、それとの一般化対相関係数を計算する。

.corr(ntop)
.rij()、.nij()、.gij()をまとめて出力。

.outlier()
外れ値の計算。テコ値の他、内部スチューデント化残差、外部スチューデント化残差、DFFITSを計算。

.enet(almax=2,almin=-4,l1r=1.0,nf=10)
SklearnによるElastic Netの計算。
almin-almax: Hyperparameterの範囲の指定。
l1r = 1.0 for LASSO, l1r=0.0 for Ridge regression
nf: fold number of cross validation

The meaning of subspace list

検出された多重共線性関係

$$\vec{x}_k = c_i^k \vec{x}_i + c_j^k \vec{x}_j$$
$$\vec{x}_l = c_i^l \vec{x}_i + c_j^l \vec{x}_j$$
$$\vec{x}_m = c_i^m \vec{x}_i + c_j^m \vec{x}_j$$

以下のようにリスト表示することにする

$$\left[\vec{x}_k, \vec{x}_l, \vec{x}_m, \left[\vec{x}_i, \vec{x}_j\right]\right]$$

Temporary basis

この場合、この5つの記述子が張る空間

$$\mathrm{span}(\vec{x}_i, \vec{x}_j, \vec{x}_k, \vec{x}_l, \vec{x}_m)$$

の基底の数は2つだけで良いことがわかる。

5つから2つの基底を自由に選んで良い（空間の持つ情報は何も失われない）

Select $\vec{x}_i, \vec{x}_j$ and remove $\vec{x}_k, \vec{x}_l, \vec{x}_m$

or

select $\vec{x}_i, \vec{x}_k$ and remove $\vec{x}_j, \vec{x}_l, \vec{x}_m$

or

select $\vec{x}_l, \vec{x}_m$ and remove $\vec{x}_i, \vec{x}_j, \vec{x}_k$

…

（選択基準は色々あるが、基本的には残したいものを残すようにすれば良い。）

このようにして、回帰精度を全く落とすことなく記述子の数を減らすことができる。

Definition of critria

TABLE I. Summary of conventional statistical criteria and newly proposed criteria for empirical-law discovery (ELD). DDR and TDR stand for descriptor-descriptor regression and target-descriptor regression, respectively.

Problem	Criterion	
	Ordinal statistics	This study (ELD)
Estimation accuracy of DDR for $\boldsymbol{x}_i$	$R_i^2 = 1 - \frac{\lvert\boldsymbol{e}_i\rvert^2}{\lvert\boldsymbol{J}_0\boldsymbol{x}_i\rvert^2}$ $\mathrm{VIF} = \frac{1}{1-R_i^2}$	$\tilde{R}_i^2 = 1 - \frac{\lvert\boldsymbol{e}_i\rvert^2}{\lvert\boldsymbol{x}_i\rvert^2}$
Importance of $\boldsymbol{x}_i$ in DDR for $\boldsymbol{x}_j$	–	$G_{ij}^2 = 1 - \frac{\lvert\boldsymbol{e}_j\rvert^2}{\lvert\boldsymbol{e}_j^{(i)}\rvert^2} = g_{ij}^2$
Pairwise correlation between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$	$r_{ij} = \frac{\boldsymbol{J}_0\boldsymbol{x}_i}{\lvert\boldsymbol{J}_0\boldsymbol{x}_i\rvert} \cdot \frac{\boldsymbol{J}_0\boldsymbol{x}_j}{\lvert\boldsymbol{J}_0\boldsymbol{x}_j\rvert}$	$g_{ij} = \frac{\boldsymbol{J}^{(i,j)}\boldsymbol{x}_i}{\lvert\boldsymbol{J}^{(i,j)}\boldsymbol{x}_i\rvert} \cdot \frac{\boldsymbol{J}^{(i,j)}\boldsymbol{x}_j}{\lvert\boldsymbol{J}^{(i,j)}\boldsymbol{x}_j\rvert}$
Estimation accuracy of TDR	$R^2 = 1 - \frac{\lvert\boldsymbol{e}_y\rvert^2}{\lvert\boldsymbol{J}_0\boldsymbol{y}\rvert^2}$	$\tilde{R}^2 = 1 - \frac{\lvert\boldsymbol{e}_y\rvert^2}{\lvert\boldsymbol{y}\rvert^2}$
Prediction capability of TDR	$Q^2 = 1 - \frac{\lvert\boldsymbol{q}_y\rvert^2}{\lvert\boldsymbol{J}_0\boldsymbol{y}\rvert^2}$ $\mathrm{AIC} = -2\ln L + 2k$	$\tilde{Q}^2 = 1 - \frac{\lvert\boldsymbol{q}_y\rvert^2}{\lvert\boldsymbol{y}\rvert^2}$
Importance of $\boldsymbol{x}_i$ in TDR	p value of T-test	$G_{iy}^2 = 1 - \frac{\lvert\boldsymbol{e}_y\rvert^2}{\lvert\boldsymbol{e}_y^{(i)}\rvert^2}$

TRi2はこれのこと

G2のこと

$$\begin{aligned}\mathrm{AIC} &= -2\ln L + 2k \\ &= m\left(\ln\left(\lvert\boldsymbol{e}_y\rvert^2\right) + \ln\left(\frac{2\pi}{m}\right) + 1\right) + 2k\end{aligned}$$

Jはrejection operator
J0は定数ベクトルからのrejection、
J(I,j)はxi、xjを除いたスパンからの
rejectionを意味する。

注意:
この右側のカラムにある指標は全て、この研究のために勝手に作ったものです。

An example of backward selection method

LIDG法では、線形独立な記述子からなる線形包（スパン）が得られ、推定精度の高いモデルが得られる。しかし、得られるモデルは一般にスパースではない。
解釈可能なスパースモデル（経験則のようなもの）を得るには、改めて記述子選択を行う必要がある。

後退選択法の指針:

1. 記述子の回帰における重要度を意味する G2 が小さい記述子を積極的に取り除くようにする。
2. 回帰係数（絶対値）がなるべく小さい記述子を取り除くようにする。
3. 除去することで、推定精度、R2、TR2があまり下がらないような記述子を除去する。
4. 除去することで、予測（汎化）精度 Q2、TQ2が上昇し、AICが下がるような記述子を除去する。

このように、各統計指標を統合的に判断し、TQ2が最大になるまで削除し続ける。
（結構、泥臭い作業となる。）
あるいは、ある指定した個数になるまで続ける。