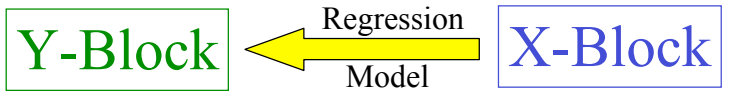
Building Predictive Models: Regression

EIGENVECTOR RESEARCH INCORPORATED

Regression



What's desired

What's measured

Regression analysis identifies the dependency between two blocks of data.

Regression models are often used to obtain estimates (or predictions) for one block of data from the other.

EIGENVECTOR RESEARCH INCORPORATED

Outline

- Nomenclature and conventions
- Classical Least Squares (CLS)
- Inverse Least Squares (ILS) Models
- Multiple Linear Regression (MLR)
- Ridge Regression (RR)
- Principal Components Regression (PCR)
- Partial Least Squares Regression (PLS)
- Determining of the Number of Factors
- Outlier Detection and Model Diagnostics
- A Unifying Theme: Continuum Regression (CR)
- Summary
- Examples

Conventions & Notation

- *Rows* correspond to *samples*, *columns* correspond to *variables*
- Notation:
  - **X** = matrix of predictor variables
  - **Y** = matrix (or vector **y**) of predicted variables
  - m = number of samples (observations)
  - nx = number of **X** variables, ny = number of **Y** variables
  - **T** = **X**-block scores matrix, $\mathbf{t}_1, \mathbf{t}_2, \ldots \mathbf{t}_k$ score vectors
  - **U** = **Y**-block scores matrix, $\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_k$ score vectors
  - **P** = **X**-block loads matrix, $\mathbf{p}_1, \mathbf{p}_2, \ldots \mathbf{p}_k$ loadings vectors
  - **Q** = **Y**-block loads matrix, $\mathbf{q}_1, \mathbf{q}_2, \ldots \mathbf{q}_k$ loadings vectors
  - **W** = **X**-block weights matrix, $\mathbf{w}_1, \mathbf{w}_2, \ldots \mathbf{w}_k$ loadings vectors
  - Θ = ridge parameter

Data Preprocessing

- Everything that was said about preprocessing for PCA goes double for regression
- Data should be linearized, if possible
- Data is almost always mean centered
- Variance scaling used when variables are in different units or greatly different magnitudes
- Outlier elimination is critical to regression models

Classical Least Squares

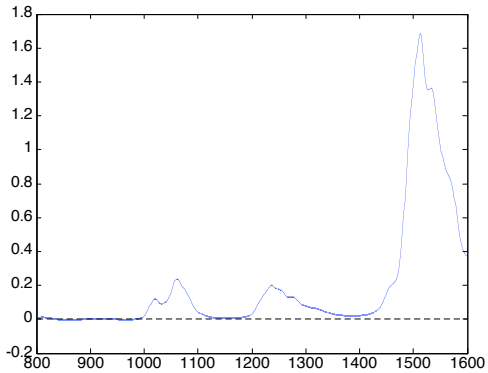
- CLS can be used to develop calibration models
  - most often used in spectroscopy
- The CLS model assumes the data follows:

$$\mathbf{X} = \mathbf{C}\mathbf{S}^{T} + \mathbf{E}$$

where **X** (mxnx) is the measured response, **S** (nxxk) is a matrix of pure component responses, **C** (mxk) is a matrix of weights (*i.e.* concentrations), and **E** (mxnx) is noise or an error matrix

The CLS Model

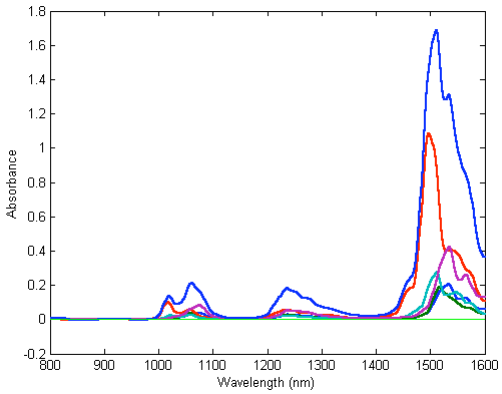
- Given known pure component spectra, how much of each does it take to make up the observed spectrum?



EIGENVECTOR RESEARCH INCORPORATED

The CLS Model

- Given known pure component spectra, how much of each does it take to make up the observed spectrum?



EIGENVECTOR RESEARCH INCORPORATED

CLS (cont.)

- Once **S** (the spectral "basis") is known, **c**, the degree to which each component contributes to a new sample **x**, can be determined from

  $\mathbf{c} = \mathbf{x}\mathbf{S}^{+}$

  where $\mathbf{S}^{+}$ is the pseudo-inverse of **S**, defined in CLS as $\mathbf{S}^{+} = \mathbf{S}(\mathbf{S}^{T}\mathbf{S})^{-1}$
- Problem: How to get **S**?

Classical Least Squares

$$\mathbf{X} = \mathbf{C}\mathbf{S}^{T} + \mathbf{E}$$
$$\mathbf{X} = \mathbf{C}\mathbf{S}^{T}$$
$$\mathbf{X}\mathbf{S} = \mathbf{C}\mathbf{S}^{T}\mathbf{S}$$
$$\mathbf{X}\mathbf{S}(\mathbf{S}^{T}\mathbf{S})^{-1} = \mathbf{C}$$
$$\mathbf{S}^{+} = \mathbf{S}(\mathbf{S}^{T}\mathbf{S})^{-1}$$

- Note that $\mathbf{S}^{T}\mathbf{S}$ is *k*x*k* (analytes by analytes) and square

Estimating S

- Sometimes, **S** can be compiled *a priori*, *e.g.* from a data base/spectral library, or from direct measurements of pure components
  - Problem: must account for all components that can contribute to **X**!
- **S** can also be estimated from mixtures, provided all **C** are known and enough samples are available:

$$\mathbf{S}^T = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{X}$$

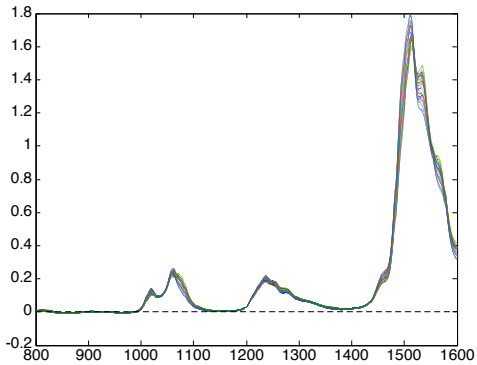
  - Problem: The concentration of *every analyte that contributes to* **X** must be known!

EIGENVECTOR RESEARCH INCORPORATED

CLS Example

- NIR data of pseudo-gasoline samples
  - absorbance at 401 channels
  - 30 samples
  - 5 analytes

```
(in analysisGUI..)

File/Load Data/Xblock:
nir_data, "spec1" array

Edit/Plot X-block (Data)
```



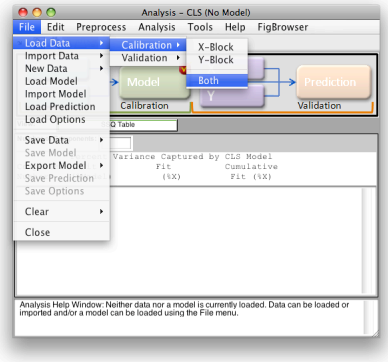
EIGENVECTOR RESEARCH INCORPORATED

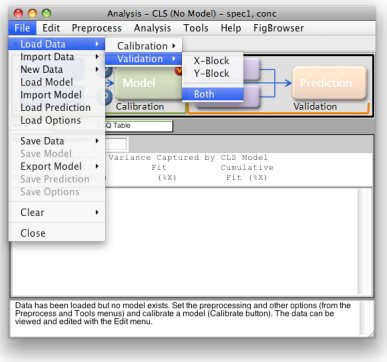
Set up Calibration and Validation Sets

```
>> load nir_data
>> conc_cal = conc(1:24,:);
>> conc_val = conc(25:30,:);
>> spec_cal = spec1(1:24,:);
>> spec_val = spec1(25:30,:);
```


EIGENVECTOR RESEARCH INCORPORATED

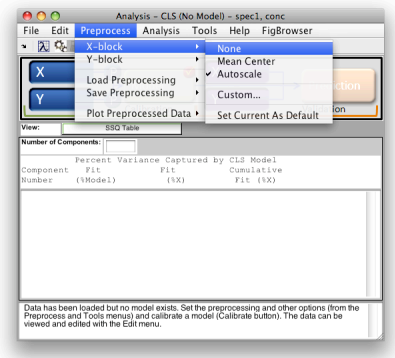
Load into Analysis Interface

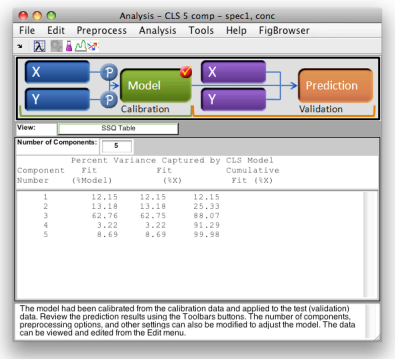




EIGENVECTOR RESEARCH INCORPORATED

Set Preprocessing, Calculate Model

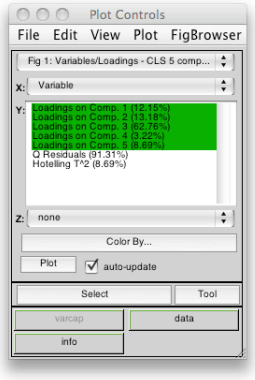


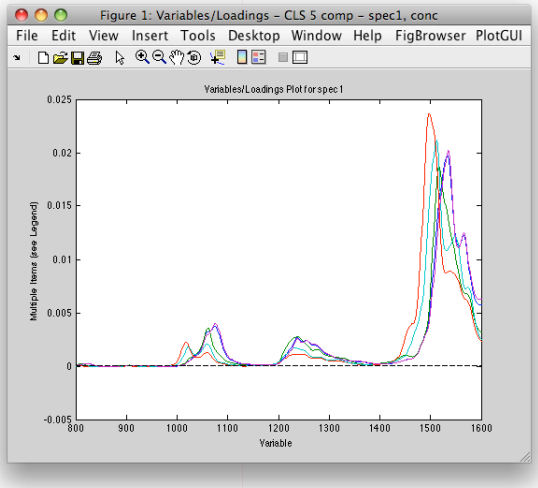


Set preprocessing to "none" in both X and Y
Click "Model" to calculate

EIGENVECTOR RESEARCH INCORPORATED

Pure Component Spectra

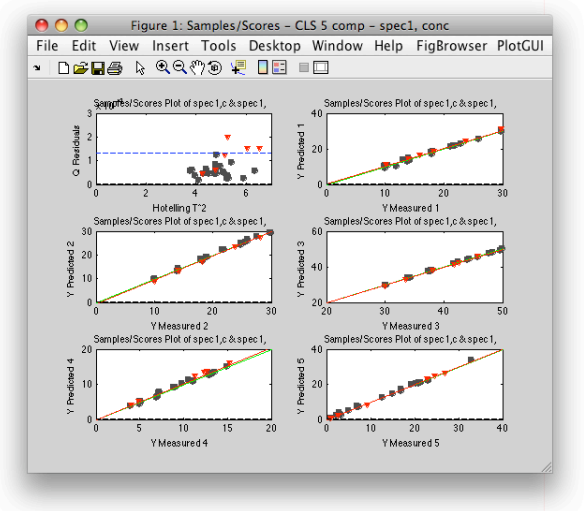




S, estimated from mixtures, using known concentrations of all 5 analytes

EIGENVECTOR RESEARCH INCORPORATED

Estimate for Unused (Test) Samples



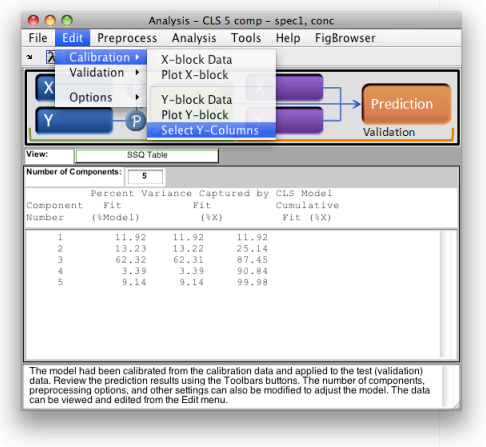
EIGENVECTOR RESEARCH INCORPORATED

CLS Problem

- What if the concentration of 1 analyte was unknown?
- Repeat the CLS procedure using only the first 4 (of 5) analytes
- Attempt to predict concentrations of unused (test) samples

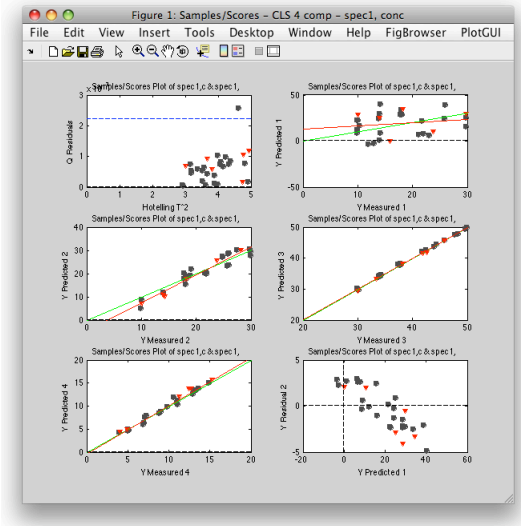
EIGENVECTOR RESEARCH INCORPORATED

Remove 5th Analyte



EIGENVECTOR RESEARCH INCORPORATED

CLS Solution-Missing Analyte



EIGENVECTOR RESEARCH INCORPORATED

Inverse Least Squares

- Inverse least squares (ILS) models assume that the model is of the form:

$$\mathbf{Xb} = \mathbf{y} + \mathbf{e}$$

where **y** (*m*x*1*) is a property to be predicted, **X** (*m*x*nx*) is the measured response, **e** (*m*x*1*) is an error vector, and **b** (*nx*x*1*) is a vector of coefficients

- Unlike CLS, ILS methods associate the noise with the predicted property, not the measured response

Estimation of *b*: MLR

- It is possible to estimate b from

$$\mathbf{b} = \mathbf{X}^{+}\mathbf{y}$$

where $\mathbf{X}^{+}$ is the pseudo-inverse of **X**

- There are many ways to obtain a pseudo-inverse most obvious is multiple linear regression (MLR), a.k.a. Ordinary Least Squares (OLS)
- In this case $\mathbf{X}^{+}$ is estimated from

$$\mathbf{X}^{+} = \left(\mathbf{X}^{T}\mathbf{X}\right)^{-1}\mathbf{X}^{T}$$

Multiple Linear Regression

$$\mathbf{Xb} = \mathbf{y} + \mathbf{e}$$
$$\mathbf{Xb} = \mathbf{y}$$
$$\mathbf{X}^T\mathbf{Xb} = \mathbf{X}^T\mathbf{y}$$
$$\mathbf{b} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$$
$$\mathbf{X}^+ = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$$

- Note that $\mathbf{X}^T\mathbf{X}$ is *nx*x*nx* and square

Advantage of ILS Methods

- ILS methods (including MLR, PCR, PLS, CR) don't require the concentration of all analytes, including interferents, be known …
- …however, interferents must vary in the calibration data set for the the ILS regression model to be robust against them

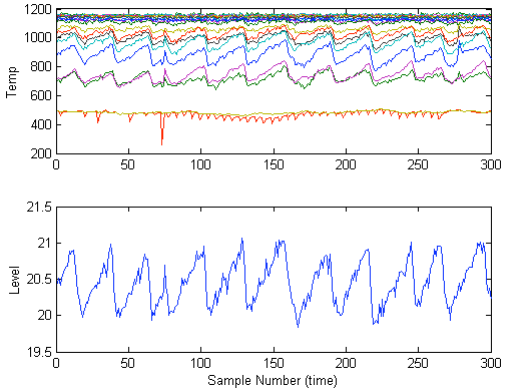
Problem with MLR

- Inverse of $\mathbf{X}^T\mathbf{X}$ only exists if
  - Rank($\mathbf{X}$) = nx, but rank($\mathbf{X}$) ≤ min (m,nx)
  - $\mathbf{X}$ has more samples than variables *i.e.* if $m \geq nx$
    - problem with spectra
  - Columns of $\mathbf{X}$ are not co-linear
- Inverse may exist but be highly unstable is $\mathbf{X}$ is nearly rank deficient
- In these cases, small perturbations in the data (possibly due to noise) can produce very different results

EIGENVECTOR RESEARCH INCORPORATED

MLR Example

- Use MLR to obtain a relationship between temperature and level in a SFCM



EIGENVECTOR RESEARCH INCORPORATED

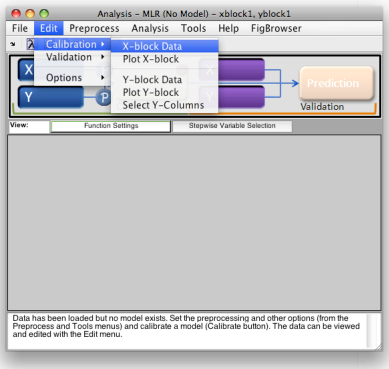
Load and Edit Data

```
>> clear
>> load plsdata
>> whos
  Name          Size            Bytes  Class

  xblock1       300x20          55240  dataset object
  xblock2       200x20          38440  dataset object
  yblock1       300x1            9008  dataset object
  yblock2       200x1            7408  dataset object

>> x = xblock1.data(delsamps([1:300]',[73 167 278 279]),:);
>> y = yblock1.data(delsamps([1:300]',[73 167 278 279]),:);
>> [mx,mnx] = mncn(x);   %center the data
>> [my,mny] = mncn(y);
```

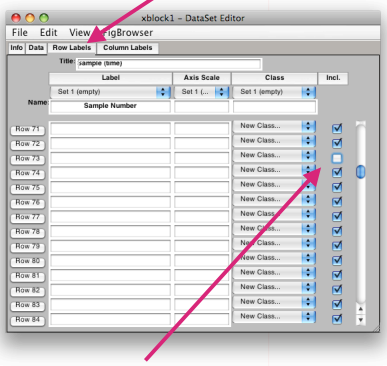

EIGENVECTOR RESEARCH INCORPORATED

...or via the Analysis GUI

1-Load x and y for both calibration and validation
2-Select Edit/Calibration/X-block Data



3-Select Row Labels tab

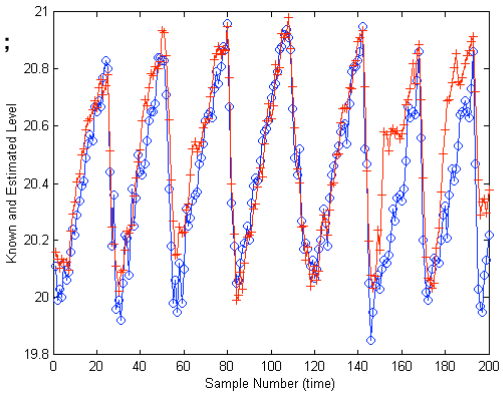


4-Deselect samples 73, 167, 278 & 279, then close

EIGENVECTOR RESEARCH INCORPORATED

MLR Regression and Results

```
>> bmlr = mx\my;
>> sx = scale(xblock2.data,mnx);
>> ymlr = rescale(sx*bmlr,mny);
```

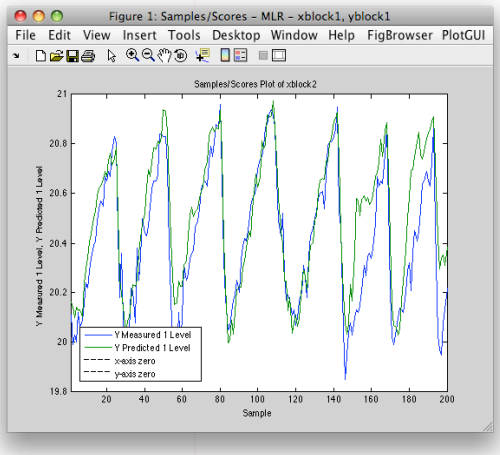



```
>> plot(1:200,yblock2.data,'o-b',1:200,ymlr,'-r+')
>> xlabel('Sample Number (time)')
>> ylabel('Known and Estimated Level')
```


EIGENVECTOR RESEARCH INCORPORATED

MLR Results

1-Set X-block preprocessing
to Mean Center
2-Click Model



3-Select Scores

EIGENVECTOR RESEARCH INCORPORATED

Ridge Regression

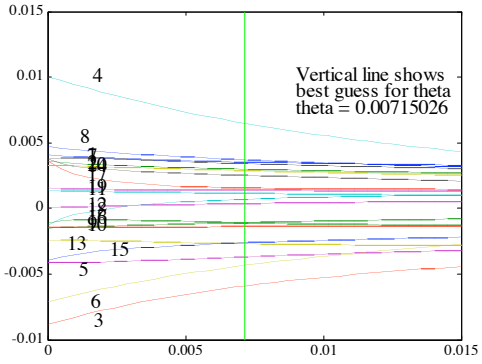
- Ridge Regression (RR) is one way to deal with ill-conditioned problems
- RR gets its name because a constant is added to the "ridge" of the covariance matrix in the formation of the pseudo-inverse:

$$\mathbf{X}^{+} = \left(\mathbf{X}^{T}\mathbf{X} + \mathbf{I}\Theta\right)^{-1}\mathbf{X}^{T}$$

- The addition of Θ stabilizes the inverse and shrinks the values of the coefficients

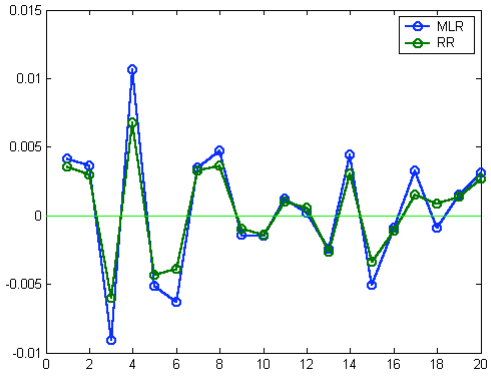
RR Shrinkage

```
[brr,theta] = ridge(mx,my,0.015,31);
```



RR and MLR Regression Vectors

```
>> plot([bmlr,brr],'o-','linewidth',2)
>> legend('MLR','RR')
>> hline
```



EIGENVECTOR RESEARCH INCORPORATED

Problem with MLR and RR

- RR helps stabilize the inverse but still has problems with strong co-linearity
- Neither MLR nor RR work when $m<nx$
- Possible solution: eliminate variables, *e.g.* stepwise regression or other variable selection
  - how to choose which variables to keep?
  - lose multivariate advantage - signal averaging
- Another solution: use PCA to reduce original variables to some smaller number of factors
  - retains multivariate advantage
  - noise reduction aspects of PCA

EIGENVECTOR RESEARCH INCORPORATED

Principal Components Regression

- Principal Components Regression (PCR) is one way to deal with ill-conditioned problems
- Property of interest **y** is regressed on PCA scores:

$$\mathbf{X}^{+} = \mathbf{P}_k \left(\mathbf{T}_k^T \mathbf{T}_k \right)^{-1} \mathbf{T}_k^T$$

- Problem is to determine *k* the number of factors to retain in the formation of the model

Principal Components Regression

$$\mathbf{T}_k \mathbf{b}_{pc} = \mathbf{y} + \mathbf{e} = \mathbf{X}\mathbf{P}_k \mathbf{b}_{pc} \Longleftrightarrow \mathbf{b} = \mathbf{P}_k \mathbf{b}_{pc}$$

$$\mathbf{T}_k \mathbf{b}_{pc} = \mathbf{y}$$

$$\mathbf{T}_k^T \mathbf{T}_k \mathbf{b}_{pc} = \mathbf{T}_k^T \mathbf{y}$$

$$\mathbf{b}_{pc} = \left(\mathbf{T}_k^T \mathbf{T}_k \right)^{-1} \mathbf{T}_k^T \mathbf{y}$$

$$\mathbf{b} = \mathbf{P}_k \left(\mathbf{T}_k^T \mathbf{T}_k \right)^{-1} \mathbf{T}_k^T \mathbf{y}$$

$$\mathbf{X}^{+} = \mathbf{P}_k \left(\mathbf{T}_k^T \mathbf{T}_k \right)^{-1} \mathbf{T}_k^T$$

- Note that $\mathbf{T}_k^T \mathbf{T}_k$ is *k*x*k* and square

Cross-Validation

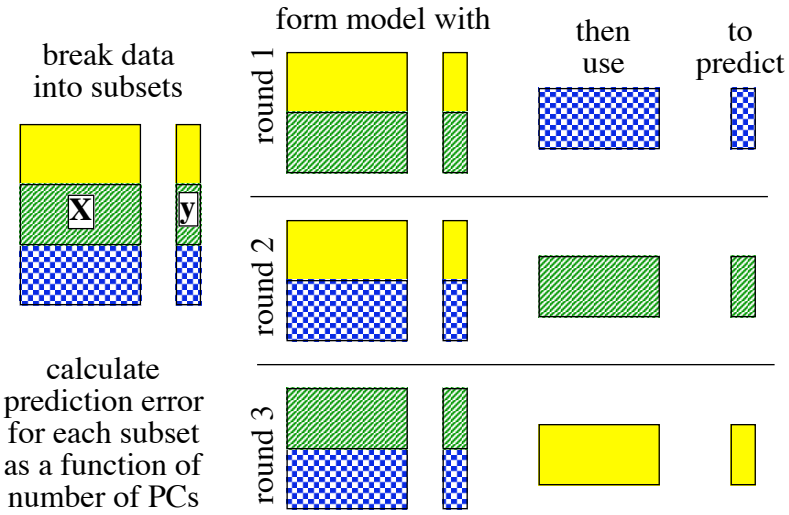
- Divide data set into *j* sample subsets
- For ***each subset*** (*j* times):
  - Build PCA model using all samples in the ***remaining*** subsets
  - Apply the model to the subset samples
  - Calculate PRESS (Predictive Residual Sum of Squares) for the subset samples:

$$\mathbf{e}^2 = \left(\mathbf{y} - \mathbf{Xb}\right)^2$$

- Look for minimum or "knee" in PRESS curve

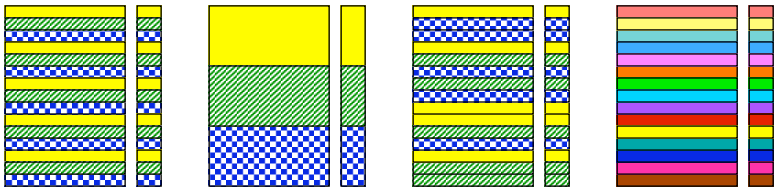
EIGENVECTOR RESEARCH INCORPORATED

Cross-validation Graphically



EIGENVECTOR RESEARCH INCORPORATED

Formation of Test Sets



"Venetian blinds" - OK when data already in random order	contiguous blocks-best for time series	random selection-usually repeated several times	leave-one-out, used when not much data available

What else? **Custom selection, based on prior knowledge!**

EIGENVECTOR RESEARCH INCORPORATED

Cross-validation Considerations

- Cross Validation method selection criteria
  - *Number* of objects in dataset
  - *Order* of objects in dataset
  - *Objective* of cross-validation (specific type of error?)
  - Presence/absence of *replicates*
- "Traps" to avoid
  - "Replicate sample trap"
    - Different replicates in both model and test set
  - "External subset selection trap"
    - Test set "space" outside of model set "space"

EIGENVECTOR RESEARCH INCORPORATED

Cross-validation Usage Matrix-I

	Venetian Blinds	Contiguous Blocks	Random Subsets	Leave-One Out	Custom
General Properties	• Easy • Relatively quick	• Easy • Relatively quick	• Easy • Can be slow, if m or number of iterations large • Selection of subsets unknown	• Easiest! (Only one parameter) • ***Avoid using if m>20***	• Flexible • Requires time to determine/ construct cross validation array
Small data sets (<~20 objects)	•	•	• OK, if many iterations done	• Good choice…. • ….unless designed/DOE data	• often needed to avoid the *external subset selection trap*
randomly-distributed objects	• Good choice	• Good choice	• Good choice • Can take a while with large m, many iterations	• OK, but…. • Can take a while with large m	•

EIGENVECTOR RESEARCH INCORPORATED

Cross-validation Usage Matrix-II

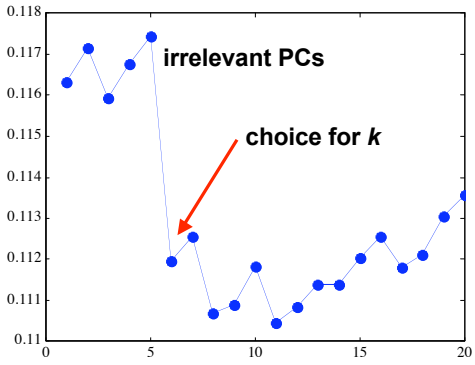
	Venetian Blinds	Contiguous Blocks	Random Subsets	Leave-One Out	Custom
time-series data	• Useful for assessing ***NON-temporal*** model errors • Can be optimistic with low number of data splits	• Useful for assessing ***temporal*** stability of model	•	•	•
Batch data	• Useful for assessing predictability ***within*** batches/parts of batches	• Useful for assessing predictability ***between*** batches/parts of batches	•	•	• Can manually select "batch-wise" test sets
Blocked data (replicates)	• Beware the *replicate sample trap* (optimistic results)!	• Good way to avoid *replicate sample trap* • Beware the *external subset selection trap* !	• Can use to avoid *replicate sample trap* (high number of splits,	• overly optimistic results, due to *replicate sample trap*	•
Designed Experiment (DOE) data	• Dangerous, unless object order is randomized	• Dangerous, unless object order is randomized	•	• Not recommended (*external subset selection trap*)	• often needed to avoid the *external subset selection trap*

EIGENVECTOR RESEARCH INCORPORATED

PCR Cross-Validation Example

- block CV since data is time series
- mean centered, 20 PCs, split 10 times

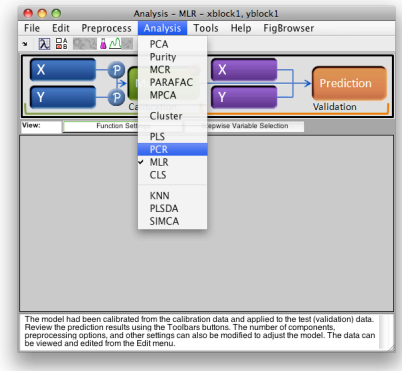
```
>> s = preprocess('meancenter');
>> opts = crossval('options');
>> opts.preprocessing = {s s};
>> [press,cumpress,rmsecv,rmsec] = crossval(x,y,'pcr',{'con',10},20,opts);
```



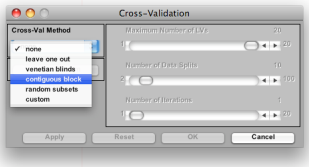
EIGENVECTOR RESEARCH INCORPORATED

...or via the GUI

1-Reset Analysis to "PCR"



2-Select Tools/Crossvalidation
3-Set to "Contiguous Block"



EIGENVECTOR RESEARCH INCORPORATED

PCR Variance Captured

```
>> options = pcr('options');
>> options.preprocessing = {s s};  % s is mean centering
>> modelpcr = pcr(x,y,6,options);
```

Percent Variance Captured by PCR Model

	X-Block		Y-Block	
PC #	This PC	Total	This PC	Total
1	81.61	81.61	85.23	85.23
2	6.16	87.77	0.19	85.41
3	5.22	92.98	0.30	85.71
4	2.54	95.53	0.02	85.74
5	1.37	96.90	0.17	85.91
6	1.01	97.91	1.09	86.99
7	0.46	98.37	0.05	87.04
8	0.39	98.76	0.27	87.31
9	0.36	99.12	0.30	87.61
10	0.24	99.37	0.02	87.63

Problems with PCR

- Some PCs not relevant for prediction, but are only relevant for describing variance in **X**
  - leads to local minima and increase in PRESS
- This is a result of PCs determined without using information about property to be predicted **y**
- A solution is to find factors using information from **y** and **X**

Partial Least Squares

- PLS is related to PCR and MLR
  - PCR captures maximum variance in X
  - MLR achieves maximum correlation between **X** and **Y**
  - PLS tries to do both by maximizing covariance between **X** and **Y**
- Requires addition of weights **W** to maintain orthogonal scores
- Factors calculated sequentially by projecting **Y** through **X**

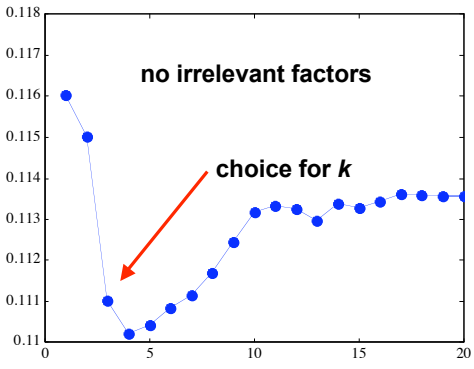
$$\mathbf{X}^{+} = \mathbf{W}_k \left(\mathbf{P}_k^{\mathrm{T}} \mathbf{W}_k\right)^{-1} \left(\mathbf{T}_k^{\mathrm{T}} \mathbf{T}_k\right)^{-1} \mathbf{T}_k^{\mathrm{T}}$$

EIGENVECTOR RESEARCH INCORPORATED

PLS Cross-Validation Example

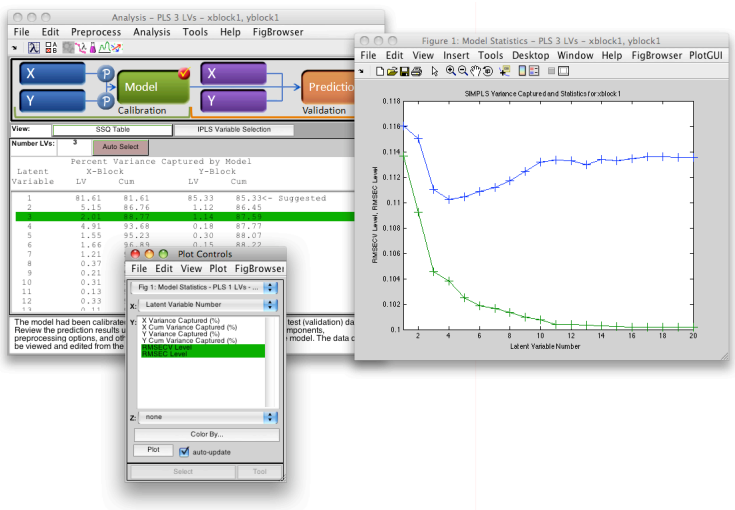
- use block CV since data is time series
- mean centered, 20 PCs, split 10 times

```
>>[press,cumpress,rmsecv,rmsec] = crossval(x,y,'sim',{'con',10},20,opts);
```



EIGENVECTOR RESEARCH INCORPORATED

...or the easy way



EIGENVECTOR RESEARCH INCORPORATED

PLS Variance Captured

```
>> options = pls('options');
>> options.preprocessing = {s s};  % s is mean centering
>> modelpls = pls(x,y,3,options);
```

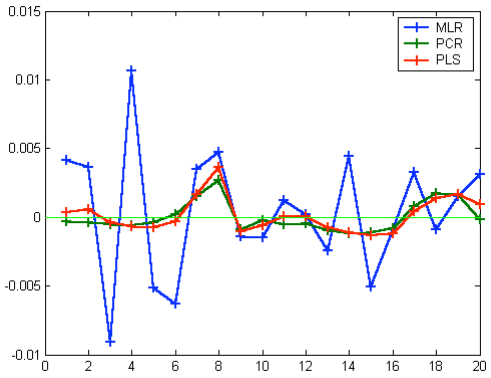
Percent Variance Captured by PLS Model

	-----X-Block-----		-----Y-Block-----	
LV #	This LV	Total	This LV	Total
1	81.61	81.61	85.33	85.33
2	5.15	86.76	1.12	86.45
3	2.01	88.77	1.14	87.59
4	4.91	93.68	0.18	87.77
5	1.55	95.23	0.30	88.07
6	1.66	96.89	0.15	88.22
7	1.21	98.10	0.05	88.27
8	0.37	98.47	0.07	88.34
9	0.21	98.68	0.08	88.42
10	0.31	99.00	0.05	88.48

EIGENVECTOR RESEARCH INCORPORATED

Regression Vectors

```
>> plot([bmlr modelpcr.reg modelpls.reg],'+-','linewidth',2)
>> legend('MLR','PCR','PLS'), hline
```



EIGENVECTOR RESEARCH INCORPORATED

PLS NIPALS Algorithm

Choose $\mathbf{u}_1 = \mathbf{y}$ or one column of $\mathbf{Y}$

$$\mathbf{w}_1 = \frac{\mathbf{X}^T\mathbf{u}_1}{\left\|\mathbf{X}^T\mathbf{u}_1\right\|} \quad (1)$$

$$\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1 \quad (2)$$

$$\mathbf{q}_1 = \frac{\mathbf{u}_1^T\mathbf{t}_1}{\left\|\mathbf{u}_1^T\mathbf{t}_1\right\|} \quad (3)$$

$$\mathbf{u}_1 = \mathbf{Y}\mathbf{q}_1 \quad (4)$$

Check for convergence by comparing $\mathbf{t}_1$ to previous $\mathbf{t}_1$. If $\mathbf{Y} = \mathbf{y}$ skip (3) and (4) and continue

$$\mathbf{p}_1 = \frac{\mathbf{X}^T\mathbf{t}_1}{\left\|\mathbf{t}_1^T\mathbf{t}_1\right\|} \quad (5)$$

$$\mathbf{p}_{1new} = \frac{\mathbf{p}_{1old}}{\left\|\mathbf{p}_{1old}\right\|} \quad (6)$$

$$\mathbf{t}_{1new} = \mathbf{t}_{1old}\left\|\mathbf{p}_{1old}\right\| \quad (7)$$

$$\mathbf{w}_{1new} = \mathbf{w}_{1old}\left\|\mathbf{p}_{1old}\right\| \quad (8)$$

Find the regression coefficient for the inner relation:

$$b_1 = \frac{\mathbf{u}_1^T\mathbf{t}_1}{\mathbf{t}_1^T\mathbf{t}_1} \quad (9)$$

After calculating scores and loadings for first Latent Variable, the X and Y-block residuals are calculated:

$$\mathbf{E}_1 = \mathbf{X} - \mathbf{t}_1\mathbf{p}_1^T \quad (10)$$

$$\mathbf{F}_1 = \mathbf{Y} - \mathbf{u}_1\mathbf{q}_1^T \quad (11)$$

Repeat entire procedure replacing $\mathbf{X}$ and $\mathbf{Y}$ with their residuals

EIGENVECTOR RESEARCH INCORPORATED

Other PLS Algorithms

- It can be shown that $\mathbf{w}_1$ is given by

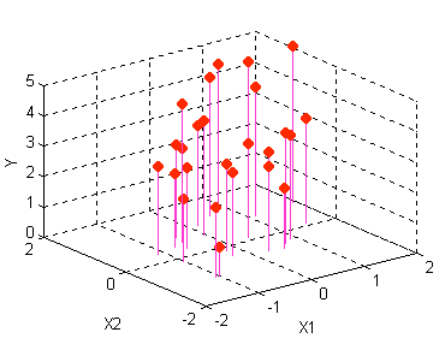
$$\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}_1 = \lambda\mathbf{w}_1$$

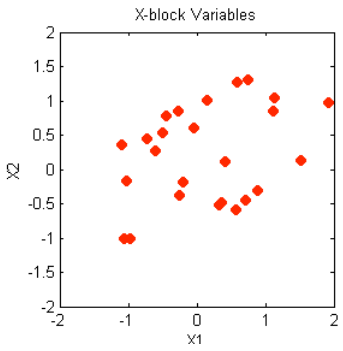
- The SIMPLS algorithm uses an orthogonalization of a Krylof sequence (faster than NIPLS algorithm)
- The important thing to remember is:

  PLS finds factors in **X** which are correlated with **Y** while describing large amounts of variance in **X**

EIGENVECTOR RESEARCH INCORPORATED

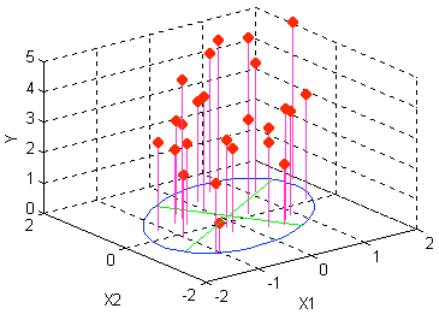
Y Projected onto X Plane

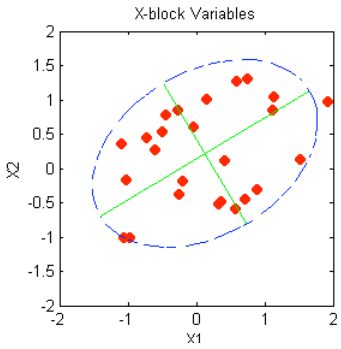




EIGENVECTOR RESEARCH INCORPORATED

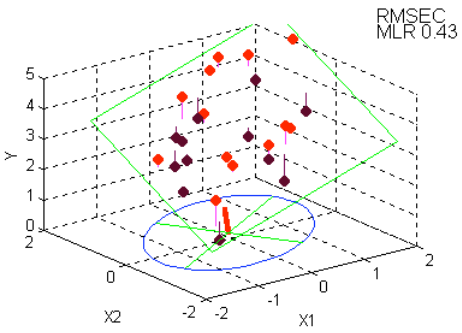
PCA of X-Block

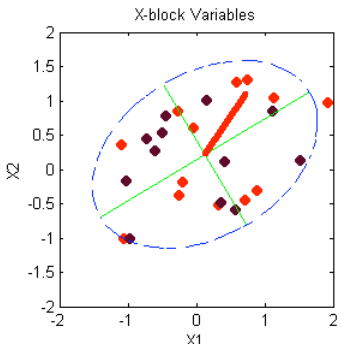




EIGENVECTOR RESEARCH INCORPORATED

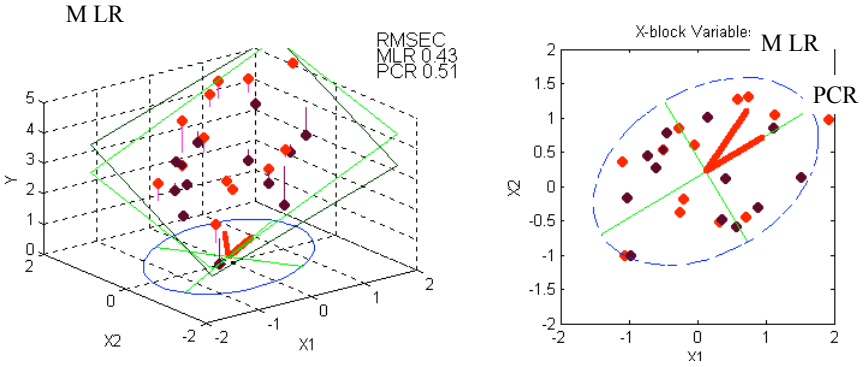
MLR Regression Vector and Surface





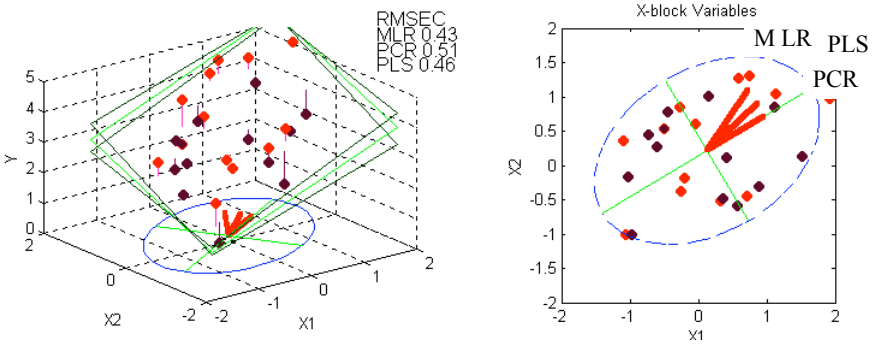
EIGENVECTOR RESEARCH INCORPORATED

PCR Regression Vector and Surface



EIGENVECTOR RESEARCH INCORPORATED

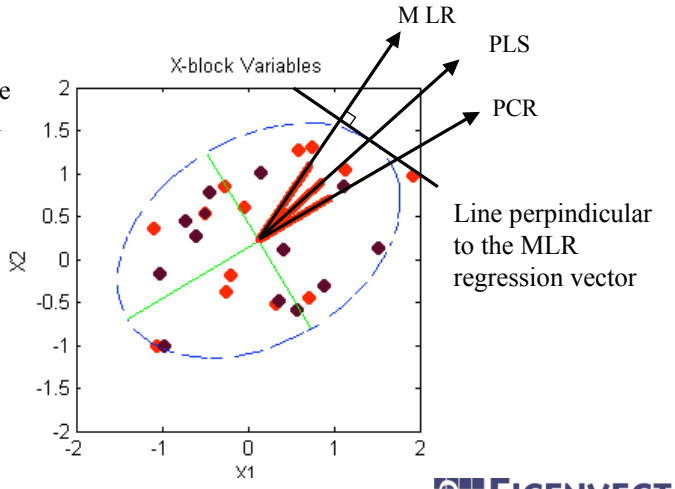
PLS Regression Vector and Surface



EIGENVECTOR RESEARCH INCORPORATED

Geometric Relationship of MLR, PCR, and PLS

PLS is the vector on the PCR ellipse upon which MLR has the longest projection



EIGENVECTOR RESEARCH INCORPORATED

PLS for Multivariate Y

- PLS can be used to relate multivariate **X** to multivariate **Y** (*a.k.a.* PLS2)
  - outer relationships

$$\mathbf{X} = \mathbf{T}_k \mathbf{P}_k^T + \mathbf{E}$$

$$\mathbf{Y} = \mathbf{U}_k \mathbf{Q}_k^T + \mathbf{F}$$

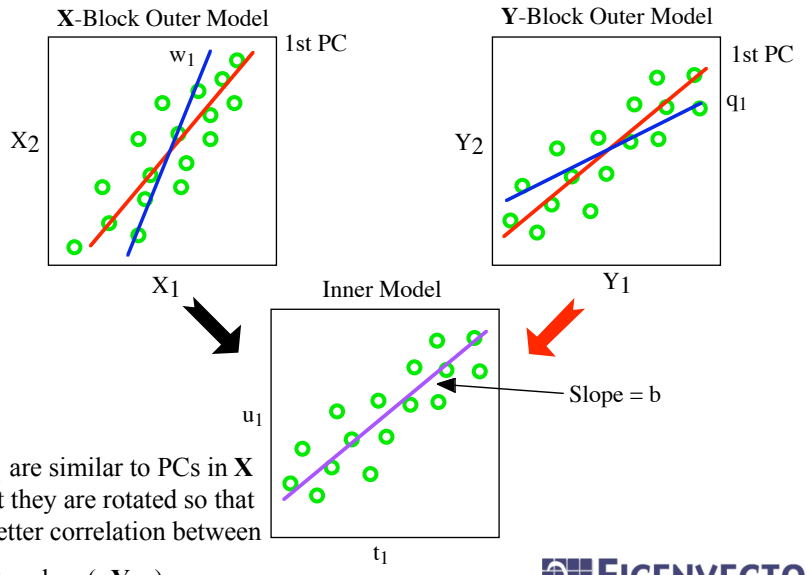
  - inner relationship

$$\mathbf{U}_k = \mathbf{T}_k \mathbf{B}_k$$

  - *i.e.* the scores in **Y** are linear combinations of the scores in **X**

EIGENVECTOR RESEARCH INCORPORATED

PLS2



$\mathbf{w}_1$ and $\mathbf{q}_1$ are similar to PCs in **X** and **Y** but they are rotated so that there is better correlation between $\mathbf{t}_1$ (=$\mathbf{Xw}_1$) and $\mathbf{u}_1$ (=$\mathbf{Yq}_1$)

EIGENVECTOR RESEARCH INCORPORATED

Model Quality Measures

- Root Mean Square Error (RMSE) Metrics
  - RMSEC
  - RMSECV
  - RMSEP
  - In units of the Y variable!

$$\sqrt{\frac{\sum_{i=1}^{m}(y_i-\hat{y}_i)^2}{m}}$$

- Correlation Coefficient (r)
  - Unit-less
  - Considers the range of Y

$$\frac{\sum_{i=1}^{m}(\hat{y}_i-\bar{\hat{y}})*(y_i-\bar{y})}{\sqrt{\left(\sum_{i=1}^{m}(\hat{y}_i-\bar{\hat{y}})*\sum_{i=1}^{m}(y_i-\bar{y})\right)}}$$

EIGENVECTOR RESEARCH INCORPORATED

Root Mean Square Error (RMSE) Metrics

- These are used to assess a model's *fit to the data* and *predictive ability on new data*
- Measures "average" deviation of model estimates from the measured data
- Measure of *fit* - root mean squared error of ***calibration*** (RMSE**C**)

$$\text{RMSEC} = \sqrt{\frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{m}}$$

*i's refer to all samples **used to build the model***

EIGENVECTOR RESEARCH INCORPORATED

Cross-Validation Error

- RMSE**C** measures *fit to the model data*. RMSECV (root mean squared error of cross-validation) is an estimate of *predictive power on new data*.
- RMSE**CV** is a function of the number of factors *k and* how the test sets were selected

$$RMSECV = \sqrt{\frac{\sum_{j=1}^{J}\sum_{i=1}^{mj}(y_i - \hat{y}_i)^2}{mj}} = \sqrt{\frac{PRESS}{mj}}$$

j's refer to different CV subsets

*i's refer to CV subset samples- **not used to build CV models***

EIGENVECTOR RESEARCH INCORPORATED

Prediction Error

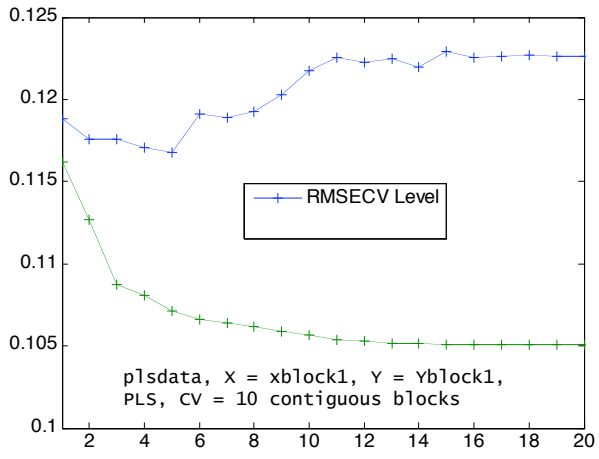
- Prediction error is often used to validate a model and is a true measure of the *predictive power on new data*
- Measure of *prediction error* - root mean squared error prediction (RMSE**P**)

$$\text{RMSEP} = \sqrt{\frac{\sum_{i=1}^{m_p}(y_i - \hat{y}_i)^2}{m_p}}$$

*i's refer to samples **NOT** used to build the model*

EIGENVECTOR RESEARCH INCORPORATED

RMSE metrics, as a function of factor (PC, LV)



RMSEC and RMSECV can also be used to determine the optimal number of factors (LVs, PCs) to be used in a model

EIGENVECTOR RESEARCH INCORPORATED

Comparison of Models

- MLR, PCR, and PLS models were constructed using SFCM data: Calibration used (xblock1) and test used (xblock2).

	MLR	PCR	PLS
RMSEC	0.0991	0.1059	0.1034
RMSECV	0.1122	0.1108	0.1098
RMSEP	0.1496	0.1366	0.1396

- Fit and prediction are two entirely different aspects of a model's performance

Number of PCs or LVs

- Choice is not always simple
- A few rules of thumb
  - sqrt(*m*) a good choice for number of splits
  - useful to do repeated CVs with different data ordering
  - if data is time series use block CV due to correlated noise
  - be conservative, models are more often overfit than underfit
  - best choice is often not the global minimum PRESS
  - look for minimum of PRESS and work backwards if improvement is not at least 2%
  - RMSEC<RMSECV by more than ~20% indicates overfit
  - look at variance captured in **X** and **Y**. Is it significant with respect to what you know about the data?