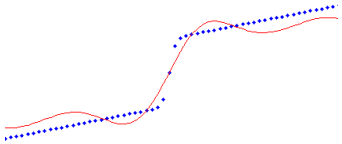
Non-linear Methods for Regression and Classification

with PLS_Toolbox and Solo



EIGENVECTOR RESEARCH INCORPORATED

Table of Contents

EIGENVECTOR RESEARCH INCORPORATED

...Continued

EIGENVECTOR RESEARCH INCORPORATED

Course Materials

- These slides
- PLS_Toolbox or Solo 8.2
- Data sets
  - From DEMS folder (distributed with software)
    - plsdata.mat (SFCM), arch.mat, paint.mat
  - From EVRIHW folder (additional data sets)
    - NL_tank_data, tecator.mat, nlmethods.mat, NIR_sugar.mat

EIGENVECTOR RESEARCH INCORPORATED

Why non-linear methods?

- Linear methods such as PCA and PLS are nice for several reasons
  1. Well-defined algorithms
  2. Reasonably fast
  3. Nice graphics
  4. Easy interpretation

- 'Smart' way to handle non-linear problems is therefore to try to turn them into linear problems

EIGENVECTOR RESEARCH INCORPORATED

Why non-linear methods?

EIGENVECTOR RESEARCH INCORPORATED

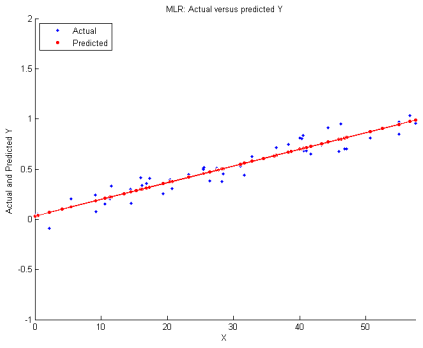
About non-linearity

- Above all:
  - Severe non-linearity is not seen in practice that often (local models)
- If the data show non-linear behaviour
  - PLS is capable of handling mild non-linearity problems
- Stronger non-linearity
  - Include cross terms of X variables
  - Modify the inner relation to e.g. a polynomial or splines
  - Focus on local linear ranges: locally weighted regression
- Serious non-linearity
  - Use more powerful tools like neural networks (take care of overfitting!)

EIGENVECTOR RESEARCH INCORPORATED

Example of linear relationship

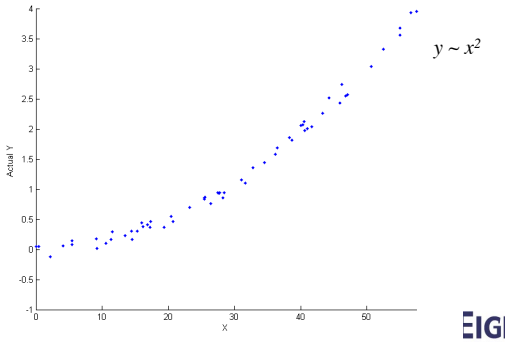
Linear method (MLR, for example) is a good model for this relationship:



EIGENVECTOR RESEARCH INCORPORATED

Example of non-linear relationship

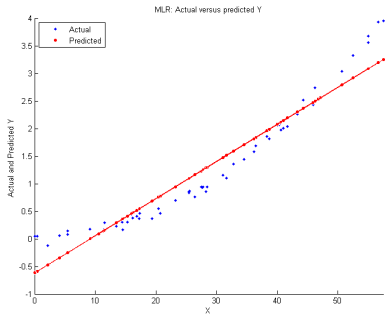
But what if the dependent variables of interest depend non-linearly on X:
Examples: $y = x^2$, $y = \log(x)$, $y = \sin(x)$, etc.



EIGENVECTOR RESEARCH INCORPORATED

Example of non-linear relationship

Linear method (MLR) is not so good when the x and y data are non-linearly related.



EIGENVECTOR RESEARCH INCORPORATED

Variable Transformations

Sometimes you can transform the x or y variable so that there is a linear relationship between the transformed x and y. Then use linear methods on the transformed data.

A. Transform the y variable

- $\boldsymbol{y^*} = G(\boldsymbol{y})$
- Apply method to $(\boldsymbol{X}, \boldsymbol{y^*})$. *E.g. MLR:* $\boldsymbol{Xb} = \boldsymbol{y^*}$
- Reverse transform to get $\boldsymbol{y} = G_{inv}(\boldsymbol{y^*})$.
- Example transformations: $log(y)$, $sqrt(y)$, $-log(I/I_0)$, etc.

Or,

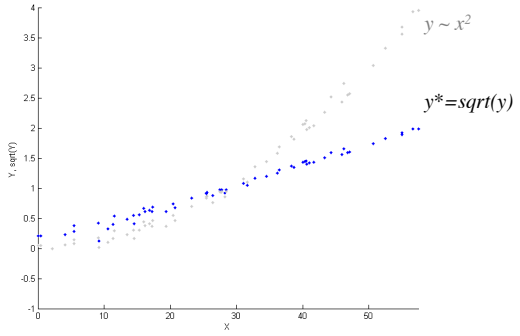
B. Transform the $\boldsymbol{X}$ variable

- or augment $\boldsymbol{X}$ with non-linear transforms

EIGENVECTOR RESEARCH INCORPORATED

Transform y

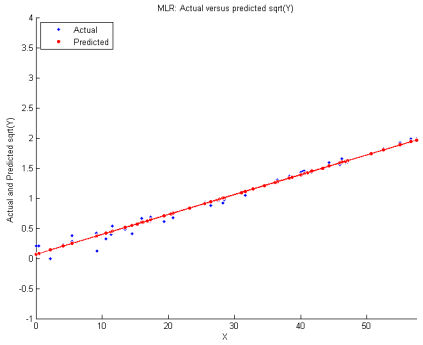
Use linear method (MLR) to predict a transformed form of y, say sqrt(y).



EIGENVECTOR RESEARCH INCORPORATED

Transform y

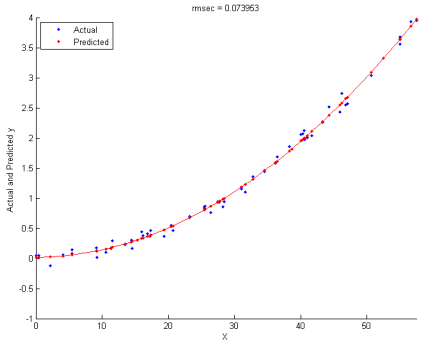
This linear method predicts the transformed *y*, *y*=sqrt(y)*, very well. Then reverse transform *y** predictions to get real predicted y.



EIGENVECTOR RESEARCH INCORPORATED

Transform y

Recovered y prediction looks good.

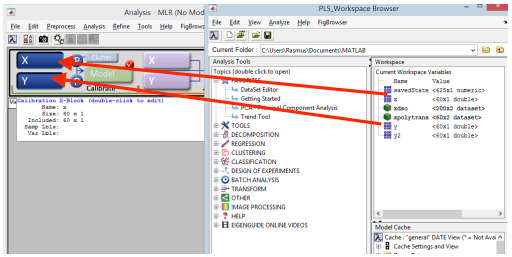


EIGENVECTOR RESEARCH INCORPORATED

Try it!

Load data into analysis window

```
>> load nlmethods.mat
>> mlr
```

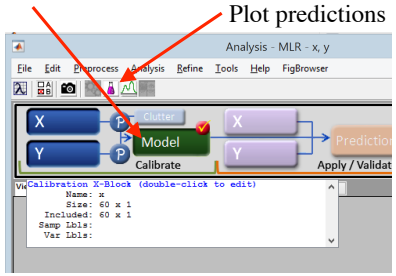


EIGENVECTOR RESEARCH INCORPORATED

Try it!

Calculate model

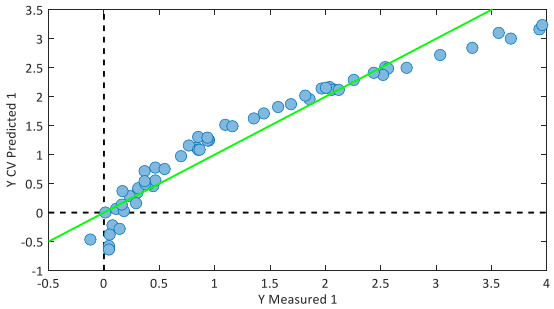
Plot predictions



EIGENVECTOR RESEARCH INCORPORATED

Try it!

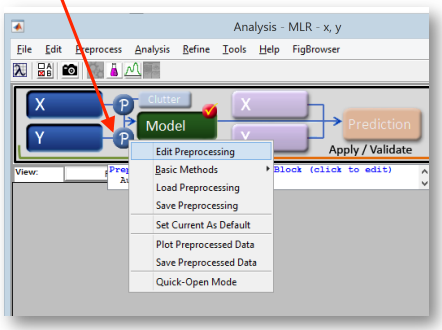
Not too impressive



EIGENVECTOR RESEARCH INCORPORATED

Try it!

Change preprocessing of y

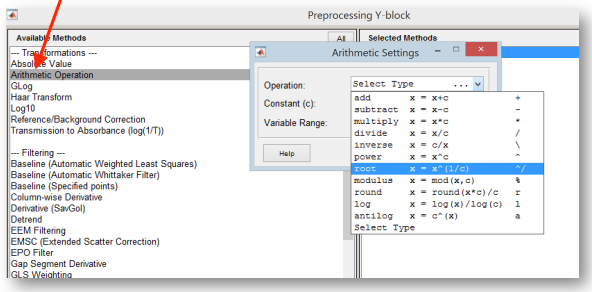


EIGENVECTOR RESEARCH INCORPORATED

Try it!

Change preprocessing of y to $y^{½}$

Do the predictions improve?

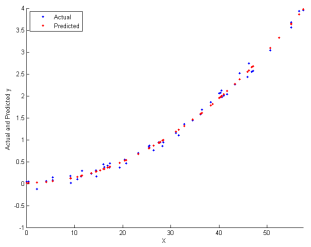


EIGENVECTOR RESEARCH INCORPORATED

Transform X and retain MLR

An alternative to transforming y is to transform x, or add new x variables which are non-linear transforms of the original x variables. Then apply linear regression using augmented x.

This is still a linear method in that the non-linearity is captured by adding non-linear variables and maintaining a linear regression model.

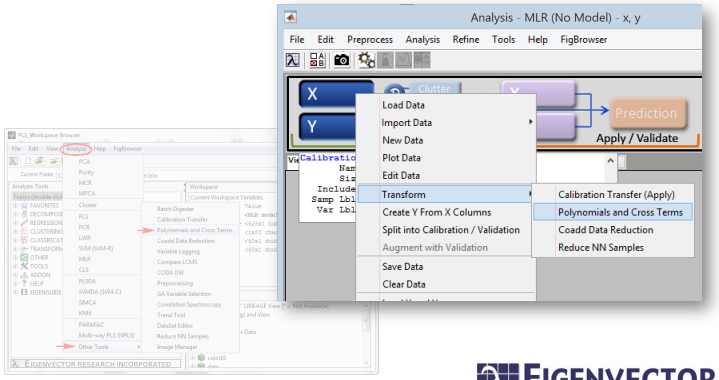


Augmenting x with squared terms allows MLR to capture the quadratic relationship between x and y.

EIGENVECTOR RESEARCH INCORPORATED

Polytransform

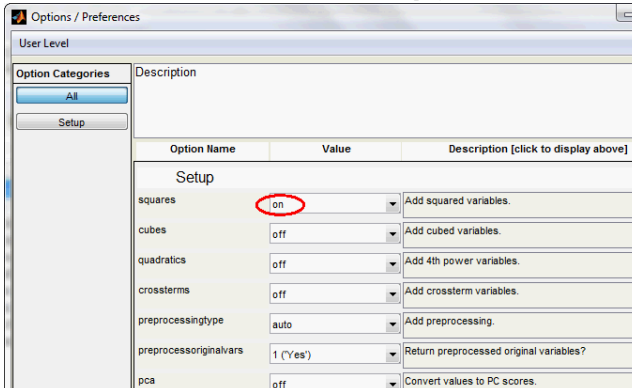
Using Browse window to augment x with squared terms or do it inside the gui:



EIGENVECTOR RESEARCH INCORPORATED

Polytransform

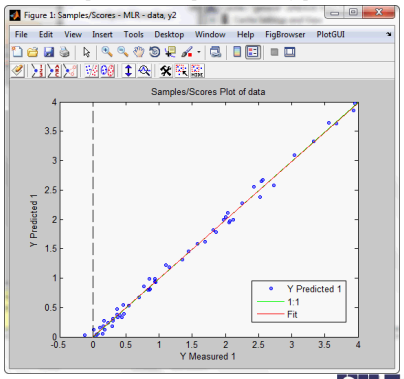
Squares = 'on', others are off. Click 'OK' and save as 'xpolytrans'.



EIGENVECTOR RESEARCH INCORPORATED

Apply MLR to polytransformed x

Use Analysis window with MLR on xpolytrans and y.
Use "Plot scores and sample statistics" plot to show y predicted versus y measured.



EIGENVECTOR RESEARCH INCORPORATED

Factor based transforms

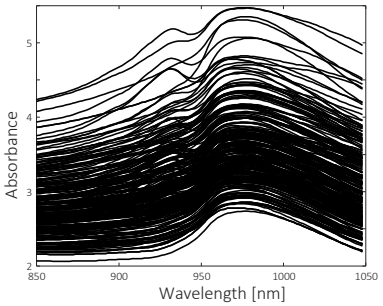
EIGENVECTOR RESEARCH INCORPORATED

Many Variable Data Sets

- Using polytransform expands the number of variables
- Especially cross terms — n variables have $n(n-1)/2$ paired cross terms. (If $n = 200$, # cross terms = 19,900)
- Increases likelihood of overfitting the data
- One solution: use PCA first then apply transform to the scores

EIGENVECTOR RESEARCH INCORPORATED

Tecator NIR calibration example



215 finely chopped pure meat samples measured by Infratec Food and Feed Analyzer.

Nonlinearity is exhibited between the spectra and the fat and moisture content. The protein content only demonstrates weak nonlinearity.

Borggaard, Thodberg, Analytical Chemistry, 64 (1992) 545–551.
Thodberg, IEEE Transactions on Neural Networks 7 (1996) 56–72.
http://lib.stat.cmu.edu/datasets/tecator

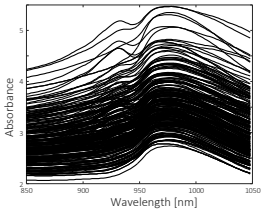
EIGENVECTOR RESEARCH INCORPORATED

Tecator: Build PLS model

Predict fat

Do first derivative and centering

Any signs of non-linearity?

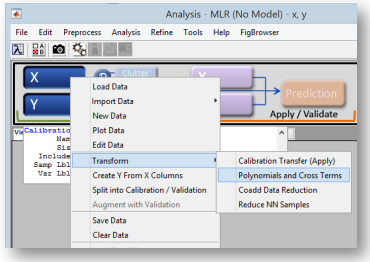


EIGENVECTOR RESEARCH INCORPORATED

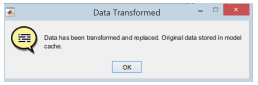
Tecator: Try a nonlinear variant

Add squares *and* crossterms. How many variables are there now?

Excessive amounts. We need a more sensible approach



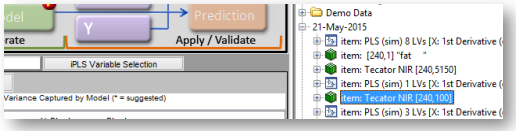
Note the message. You will need it to get the original data back



EIGENVECTOR RESEARCH INCORPORATED

Tecator: Try a nonlinear variant

Double click here to get original data back



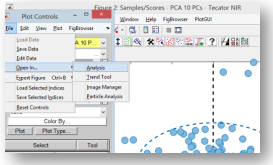
Build a PCA model with enough components*

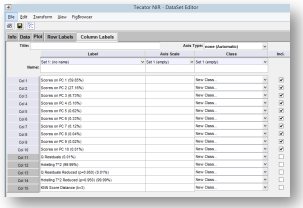
* Not important to use the *right* number, just make sure all relevant variation is included – be optimistic!

EIGENVECTOR RESEARCH INCORPORATED

Tecator: Extract scores

Make a score plot and open scores in Analysis





Make sure that only scores are selected. In fact, hard delete the rest.

Do squares and interactions

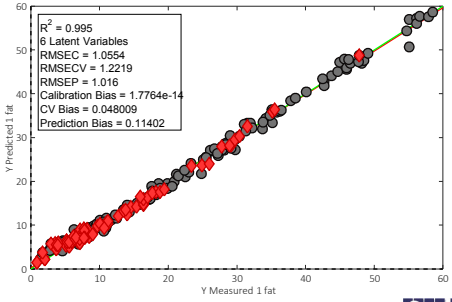
What preprocessing?

EIGENVECTOR RESEARCH INCORPORATED

Tecator: Is the PLS model nice?

If not, then try to see if variable selection can improve

Possible set aside a test set



EIGENVECTOR RESEARCH INCORPORATED

Tecator: Overfit

Try using just 20% of the data for calibration and ten PC's. How does that work?

EIGENVECTOR RESEARCH INCORPORATED

Locally Weighted Regression

EIGENVECTOR RESEARCH INCORPORATED

Locally Weighted Regression

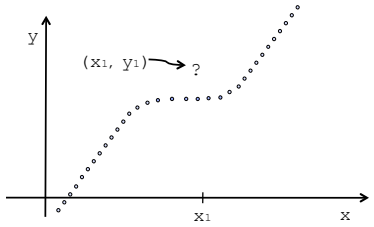
Nonlinear relations can often be approximated by a linear function on a small (local) scale.

LWR models work by choosing a subset of the calibration data (the "local" calibration samples) to create a "local" model for a given new sample. The local calibration samples are identified as the samples closest to a new sample.

EIGENVECTOR RESEARCH INCORPORATED

LWR

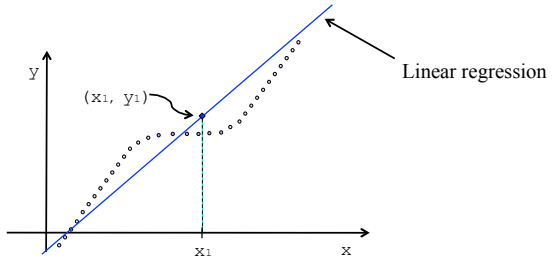
1-D example showing advantage of locally weighted regression over linear regression. Predict y for a new x value.



EIGENVECTOR RESEARCH INCORPORATED

LWR

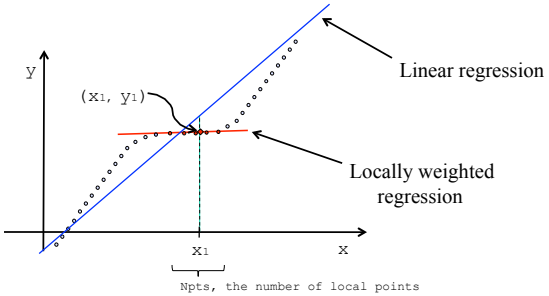
Linear regression is good if there is a simple linear relationship between y and x.



EIGENVECTOR RESEARCH INCORPORATED

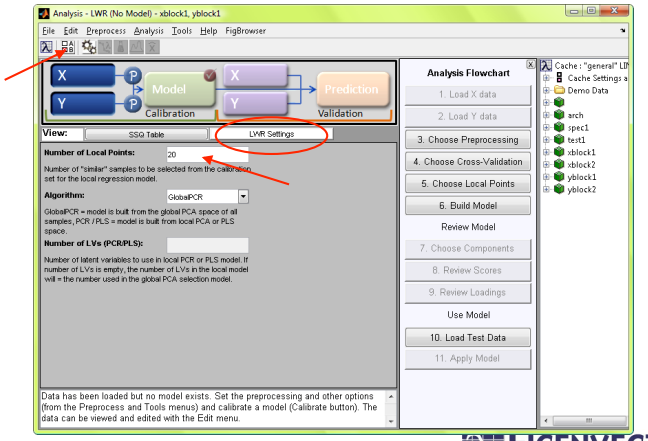
LWR

Advantage of locally weighted regression over linear regression for more complex relationship between y and x.



EIGENVECTOR RESEARCH INCORPORATED

Using LWR from the GUI.



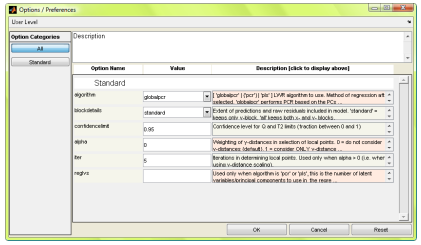
EIGENVECTOR RESEARCH INCORPORATED

More advanced options

alpha: Weighting of y-distances in selection of local points.

0 = do not consider y-distances {default},
1 = consider ONLY y-distances.

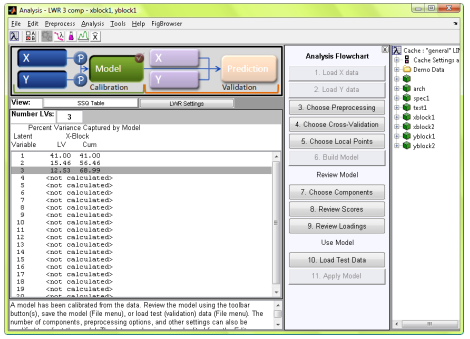
any non-zero alpha = select samples which are close in both the PC space and have similar y-values.



EIGENVECTOR RESEARCH INCORPORATED

Build model

Try on Tecator. GlobalPCR or PLS works best?



EIGENVECTOR RESEARCH INCORPORATED

LWR References

- Wiki: `http://wiki.eigenvector.com/index.php?title=Lwr`
- Naes, T., T. Isaksson, and B. Kowalski, (1990). Locally weighted regression and scatter correction for near-infrared reflectance data. Anal. Chem., 1990, 62 (7), pp 664–673.
- Wang, Z., T. Isaksson, B. R. Kowalski, (1994). New approach for distance measurement in locally weighted regression. Anal. Chem., 1994, 66 (2), pp 249–260.

EIGENVECTOR RESEARCH INCORPORATED

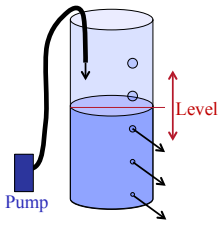
Hierarchical Models

- Sort of a "Manual" Locally Weighted Regression
- Calibration:
  1. Build top-level linear model for estimating
  2. Identify sub-regions of data which are roughly linear
  3. Build separate (linear) sub-models on sub-regions for more accurate estimates in sub-ranges
  4. Build Hierarchical model which selects appropriate sub-model based on top-level model estimate
- Example: NL_tank_data

EIGENVECTOR RESEARCH INCORPORATED

NL_Tank_Data

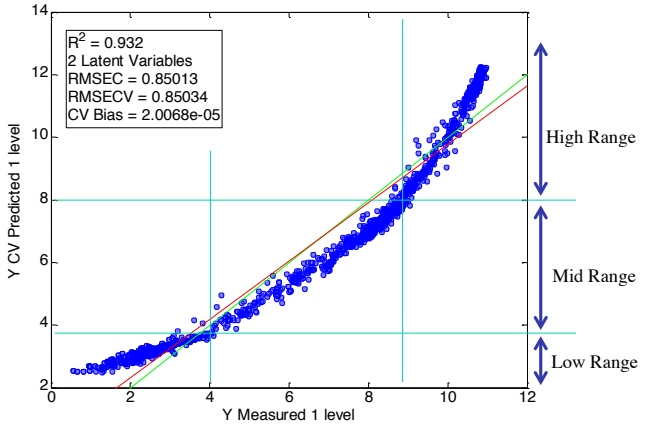
- Non-linear tank level control experiment
- Input is the voltage to a pump which fills a tank. The tank has numerous outlet holes, so is somewhat more complicated than a single hole
- Each line of the calnu data (X) contains the last 6 pump inputs (in pulse form) for the corresponding level in calny (Y)



(hole sizes are not accurate as to experimental setup!)

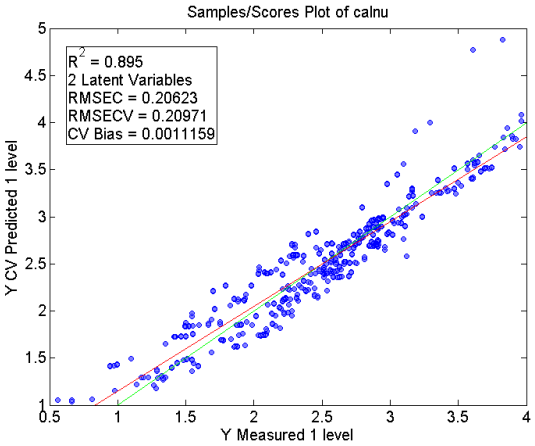
EIGENVECTOR RESEARCH INCORPORATED

Global PLS Model Results

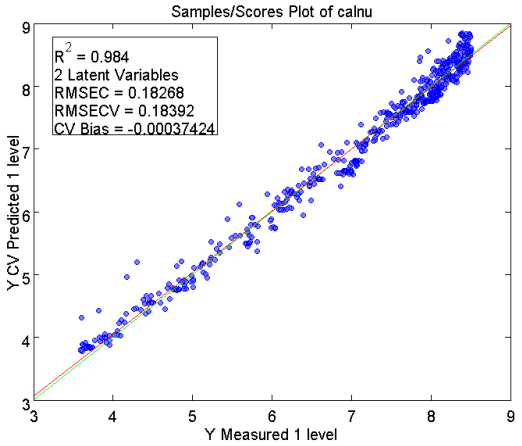


EIGENVECTOR RESEARCH INCORPORATED

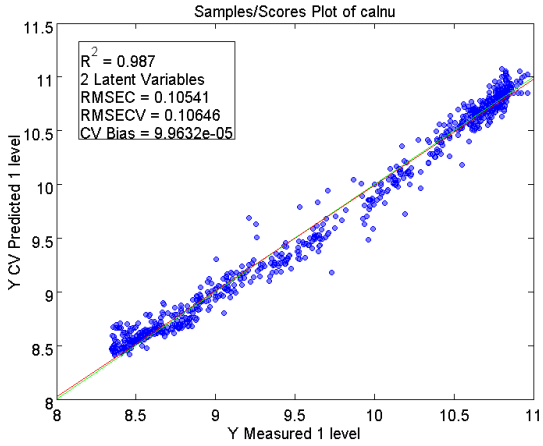
Low-Range Model Results



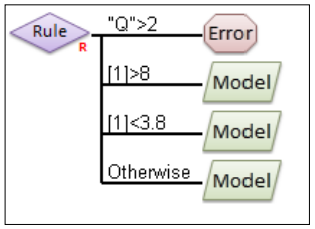
Mid-Range Model Results



High-Range Model Results



Single Layer Hierarchical Model



If Q is too large, throw error

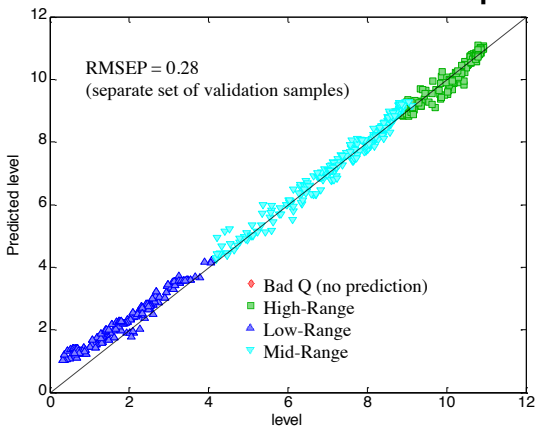
If Predicted Y is > 8, apply "High-Range" model

If Predicted Y is < 3.8, apply "Low-Range" model

Otherwise, apply "Mid-Range" model

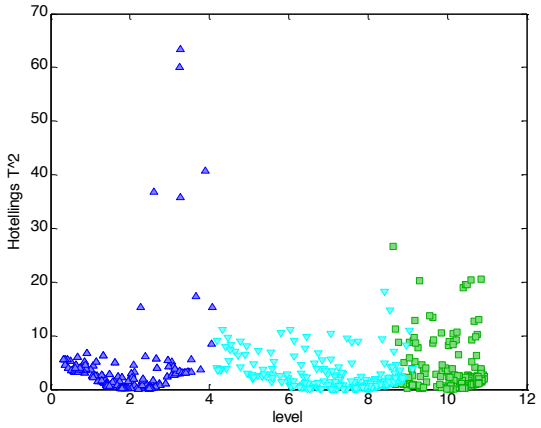
EIGENVECTOR RESEARCH INCORPORATED

Hierarchical Model Output



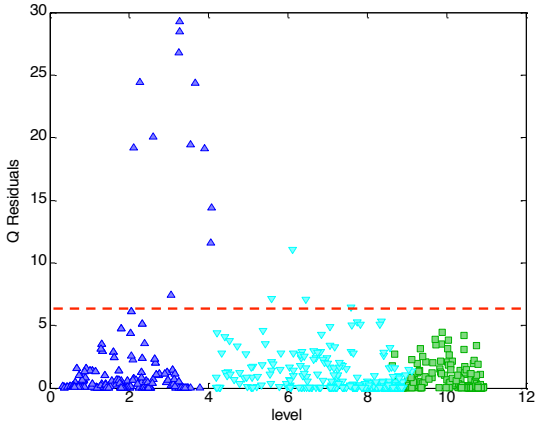
EIGENVECTOR RESEARCH INCORPORATED

Hierarchical Model Output



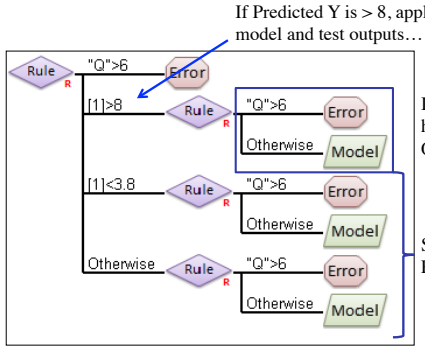
EIGENVECTOR RESEARCH INCORPORATED

Hierarchical Model Output



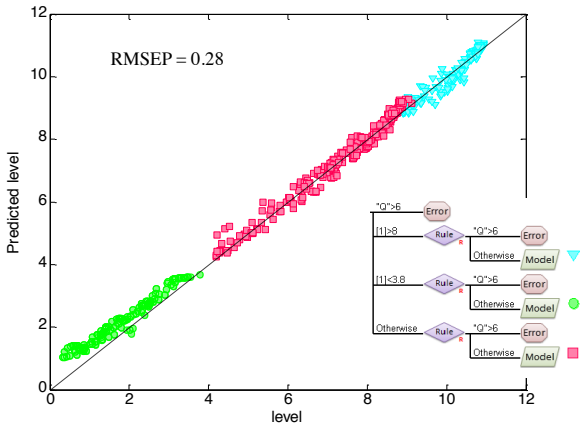
EIGENVECTOR RESEARCH INCORPORATED

Add Layer of Output Testing



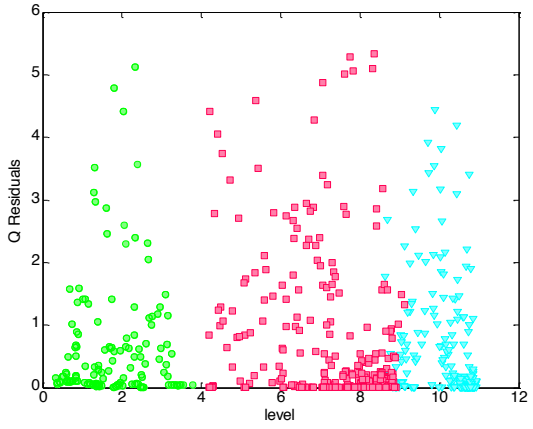
EIGENVECTOR RESEARCH INCORPORATED

2-Layer Hierarchical Model Output



EIGENVECTOR RESEARCH INCORPORATED

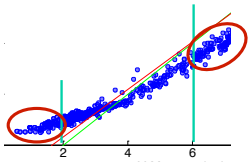
2-Layer Hierarchical Model Output



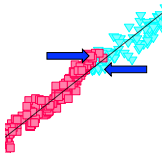
EIGENVECTOR RESEARCH INCORPORATED

Considerations

- Sub-models should include samples from "outside" the range they will be used in



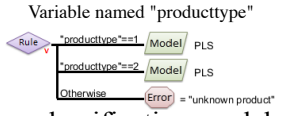
- Anticipate "transition" effects. Step effects may be observed between models. Avoid putting transitions in critical places.



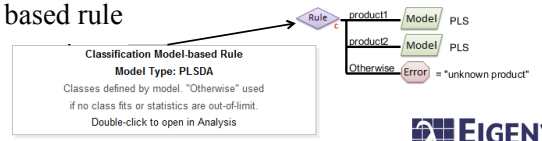
EIGENVECTOR RESEARCH INCORPORATED

Classification-Based Selection of Regression Models

- Many systems have "domains" which are best fit by individual models
- Domain indicated by indicator variable = Variable-Based rule



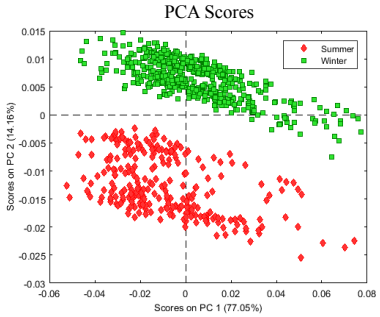
- Domain inferred from data = classification-model-based rule



EIGENVECTOR RESEARCH INCORPORATED

Diesel Fuel BP50 Determination

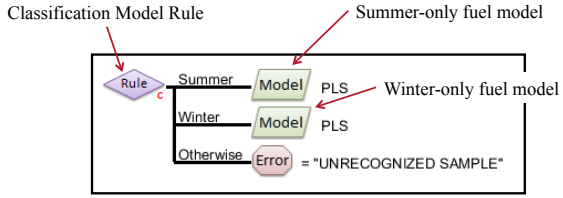
- BP50 = boiling point at 50% recovery (deg C, ASTM D 86)
- Two versions of the fuel, Winter and Summer
- NIR Spectra
- SWRI_Diesel_NIR.mat



- http://www.eigenvector.com/data/SWRI/

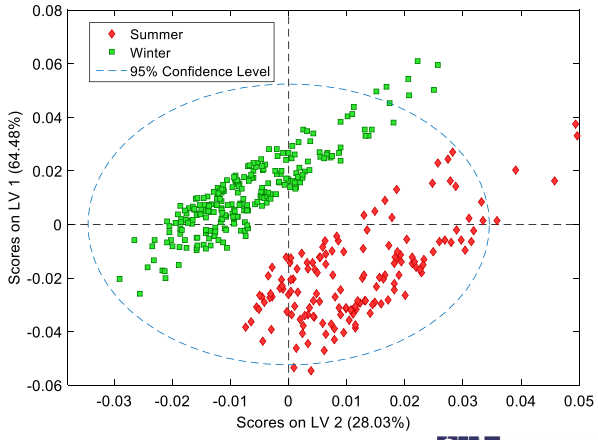
EIGENVECTOR RESEARCH INCORPORATED

Local Regression Model Based on NIR Spectra of Diesel Fuel



EIGENVECTOR RESEARCH INCORPORATED

PLSDA Classification Model



EIGENVECTOR RESEARCH INCORPORATED

Hands-On

- Using SWRI Diesel Data (in homework folder)
- Preprocessing: 1st Derivative + Mean Centering
- Two PLS models: Summer-Only regression model and a Winter-Only regression model (predicting first column of y-block, BP50)
- PLSDA Model: Summer vs. Winter
- Assemble Hierarchical Model
- Output: BP50 prediction and Q residuals

EIGENVECTOR RESEARCH INCORPORATED

Support Vector Machines

Support Vector Machines (SVMs) are a set of related supervised learning techniques for **classification** and **regression** which became popular over the past decade.

EIGENVECTOR RESEARCH INCORPORATED

Support Vector Machines

- Since version 5.8, PLS_Toolbox provides an interface to the commonly-used and freely available "LIBSVM" implementation (version 2.9) by Chang and Lin. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
- Wrapped for ease of use:
  Calibration and Prediction using same function
  Automatic parameter selection
  Standard PLS_Toolbox syntax including options and default values
- Regression (SVM) and Classification (SVMDA) support, command line and Analysis window usage.
- Low-level access to LIBSVM functions if desired!

EIGENVECTOR RESEARCH INCORPORATED

SVM Outline

Introduce Support Vector Machines (SVMs) as binary linear classifier

- Decision boundary between two classes in X-space
- Extensions to multi-class
- Extensions to handle non-separable problems ("cost", "gamma" parameters)
- "Nu" parameter is an alternative to "cost".
- How to find the best parameters to use

Using SVM-Classification in Analysis Window and command line

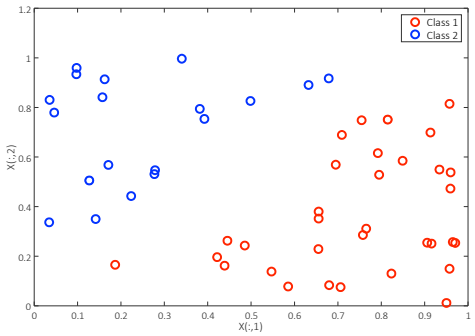
Extension to SVM regression

- "Epsilon" parameter

EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

SVMs finds the optimal separating margin between each pair of classes.

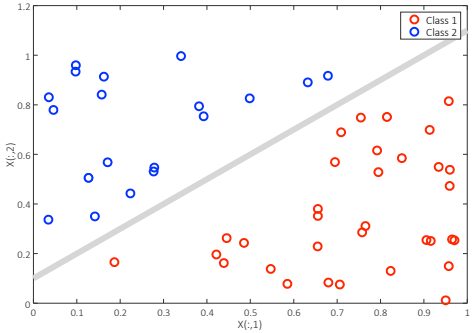


Example: samples belong to one of two classes, A and B, in two variable space (x1, x2).

EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

SVMs finds the optimal separating margin between each pair of classes. $0 = \mathbf{x}^T\mathbf{w} + b$

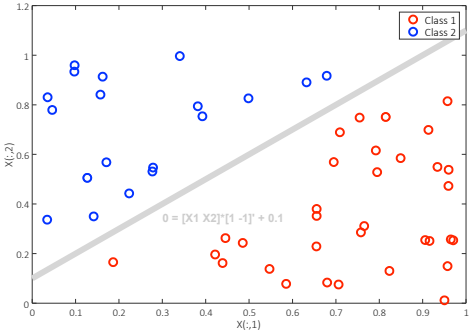


Example: samples belong to one of two classes, A and B, in two variable space (x1, x2).

EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

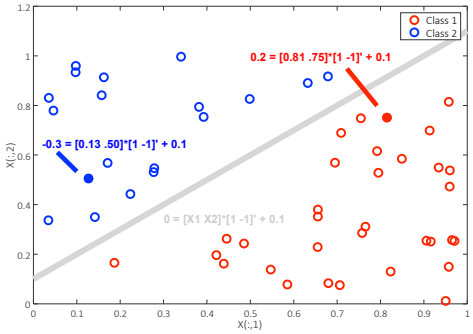
SVMs finds the optimal separating margin between each pair of classes. $0 = \mathbf{x}^T\mathbf{w} + b$



EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

If the point has a positive value, it is red and if negative blue

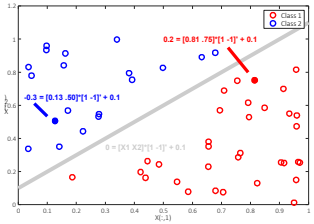


EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

Find 'regression' vector **w** such that inner product of sample $\mathbf{x}_i$ ($+b$) will have the right sign: y_i is either +1 or -1.

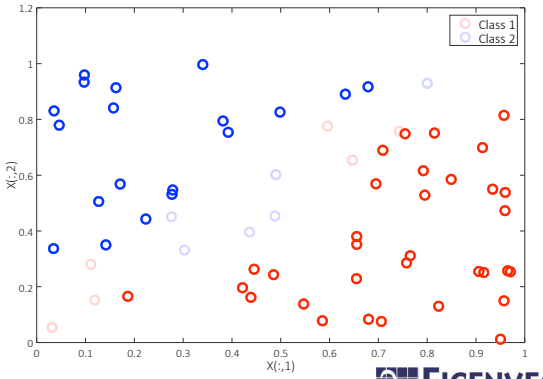
$$y_i = \text{sign}[\mathbf{w}^T\mathbf{x}_i + b]$$



EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

But what if a nice line cannot be found?



EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

First, let's reformulate the linear SVM. Instead of just solving

$$y_i = \text{sign}[\mathbf{w}^T\mathbf{x}_i + b]$$

We do

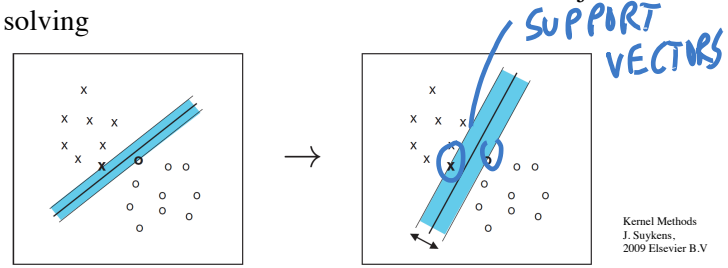
$$\min(\mathbf{w}^T\mathbf{w}) \quad \text{subject to} \quad y_i([\mathbf{w}^T\mathbf{x}_i + b]) \geq 1$$

This will aim to find a classifier that works (!) and has a wide margin (the margin equals $2/\|\mathbf{w}\|$)

EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

First, lets reformulate the linear SVM. Instead of just solving



Kernel Methods
J. Suykens,
2009 Elsevier B.V

This will aim to find a classifier that works (!) and has a wide margin (the margin equals $2/\|\mathbf{w}\|$)

EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

First, lets reformulate the linear SVM. Instead of just solving

$$y_i = \text{sign}[\mathbf{w}^T\mathbf{x}_i + b]$$

We do

$$\min(\mathbf{w}^T\mathbf{w})½ \quad \text{subject to} \quad y_i([\mathbf{w}^T\mathbf{x}_i + b]) \geq 1$$

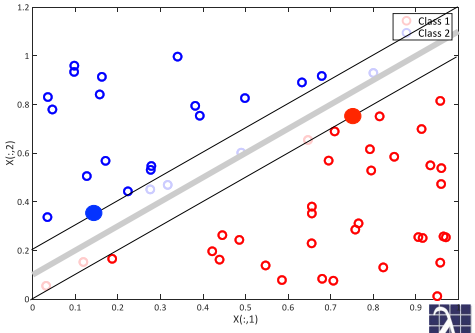
This will aim to find a classifier that works (!) and has a wide margin (the margin equals $2/\|\mathbf{w}\|$)

EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

$\min(\mathbf{w}^T\mathbf{w})$ subject to $y_i([\mathbf{w}^T\mathbf{x}_i+b])\geq 1$

Support vectors = the ones where the equality holds. The ones further out don't matter, once **w** and *b* are found

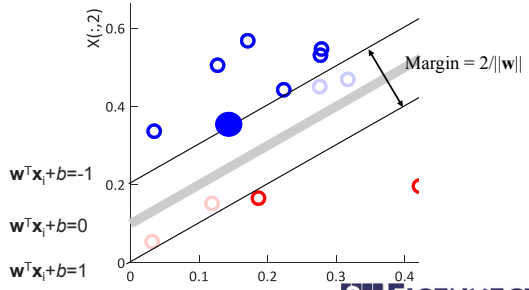


EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

$\min(\mathbf{w}^T\mathbf{w})$ subject to $y_i([\mathbf{w}^T\mathbf{x}_i+b])\geq 1$

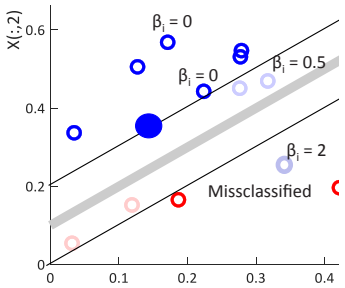
Support vectors = the ones where the equality holds. The ones further out don't matter, once **w** and *b* are found



EIGENVECTOR RESEARCH INCORPORATED

Allowing misclassification

$\min(\mathbf{w}^T\mathbf{w})+C\Sigma\beta_i$ subject to $y_i([\mathbf{w}^T\mathbf{x}_i+b])\geq(1-\beta_i)$



Cost given by *C*.

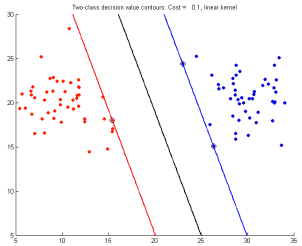
When zero, don't worry about misclassifications,

When big (up to infinity), no errors allower (=smaller margin)

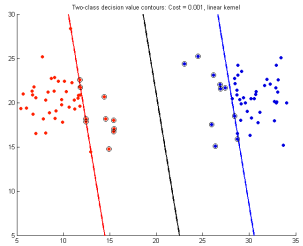
EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

Cost: (0 – infinity). When high, fewer samples within narrower margin, less misclassification, maybe overfitting.



cost = 0.1



cost = 0.001

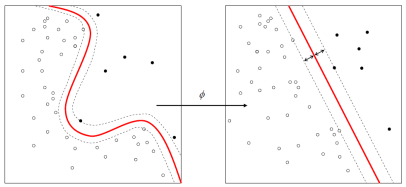
EIGENVECTOR RESEARCH INCORPORATED

SVM nonlinearity

The Gaussian RBF kernel function takes the following form:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}$$

gamma: (0 – infinity). Low, linear; high local and nonlinear



https://en.wikipedia.org/wiki/Kernel_method

EIGENVECTOR RESEARCH INCORPORATED

SVM Parameters

SVM classification involves defining parameters (**cost**, **gamma**).

Cost: (0 – infinity). When high, allow less misclassification but could cause overfitting.

gamma: (0 – infinity). Low, linear; high local and nonlinear

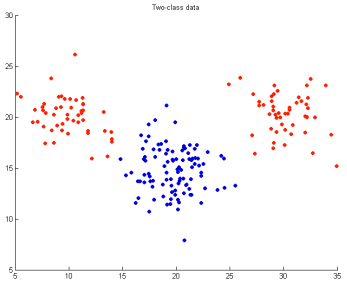
The svm function and GUI selects automatically by default using cross-validation.

EIGENVECTOR RESEARCH INCORPORATED

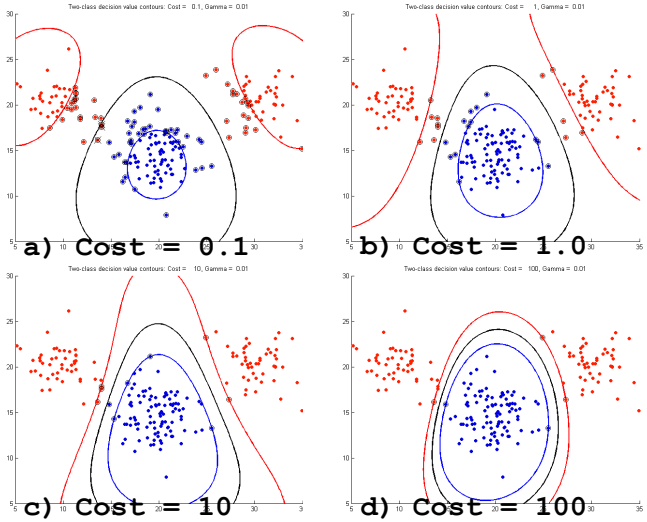
SVM Parameters

Examine the effect of the cost/nu and gamma parameters using simple two-variable, two-class dataset with 100 red and 100 blue data points.

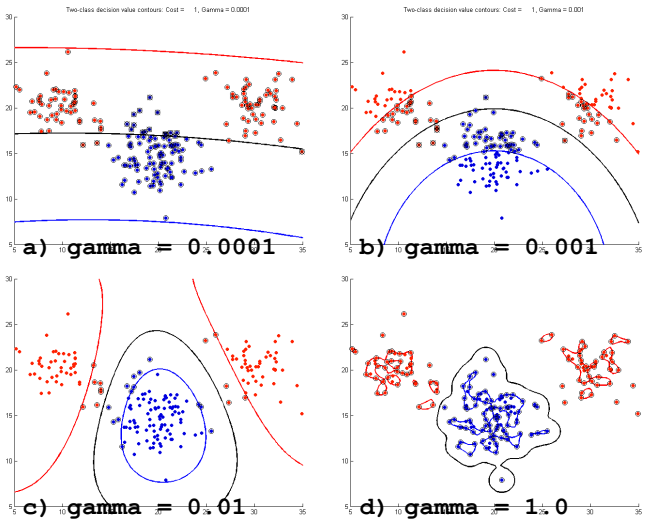
These two classes are not linearly separable but are not too complicated.



EIGENVECTOR RESEARCH INCORPORATED



RESEARCH INCORPORATED



Summary of parameter effects

Increase cost (decrease nu)

Narrower separating margin and fewer support vectors.

Increase gamma

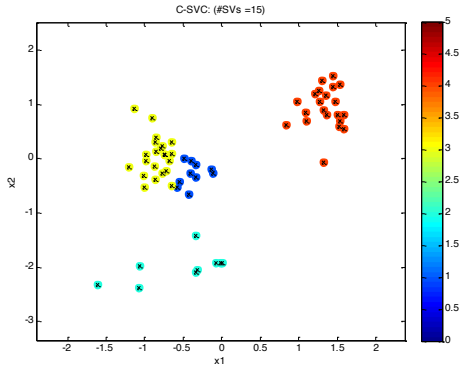
More complicated decision boundary shape. Very small gamma gives linear kernel behavior (decision boundary is a plane).

For more detailed discussion of SVM parameters see http://wiki.eigenvector.com/index.php?title=Svmda.

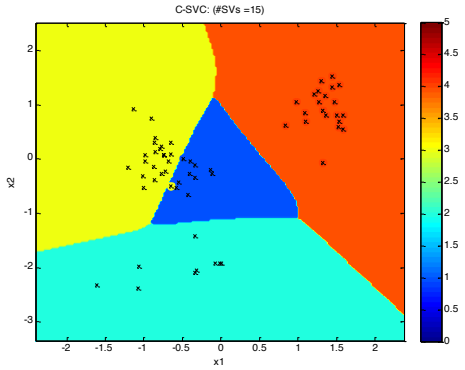
Multi-class SVM Classification

- Classifying data belonging to more than two classes (k > 2) is handled by considering each pair of classes as a separate SVM problem. Hence k*(k-1)/2 SVM classifiers
- LIBSVM implements the "one-against-one" approach for multiclass classification. If k is the number of classes, then k(k - 1)/2 classifiers are constructed and each one trains data from two classes. In classification of a new sample we use a voting strategy: each binary classification is considered to be a voting where votes can be cast for the class of the new sample. In the end a point is designated to be in a class with the maximum number of votes.

Multi-class SVM Classification



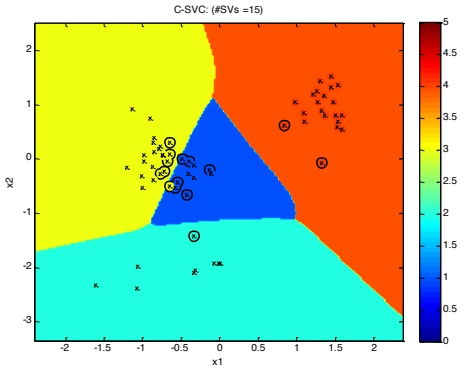
Multi-class SVM Classification



New test dataset samples are assigned to a class according to which partition they reside in.

EIGENVECTOR RESEARCH INCORPORATED

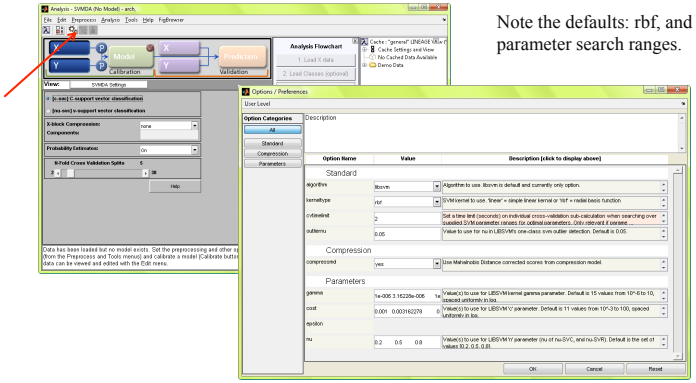
Multi-class SVM Classification



o = Support vectors

EIGENVECTOR RESEARCH INCORPORATED

SVM C-classification using the PLS_toolbox GUI

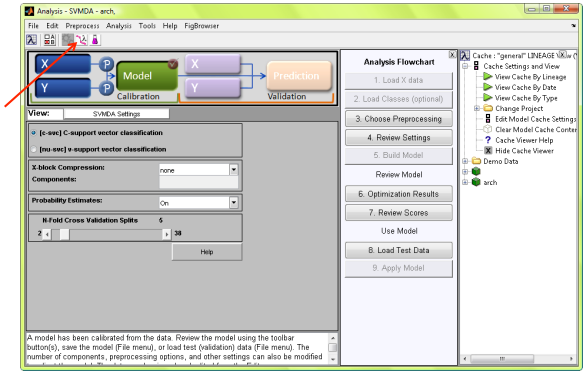


Note the defaults: rbf, and parameter search ranges.

EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

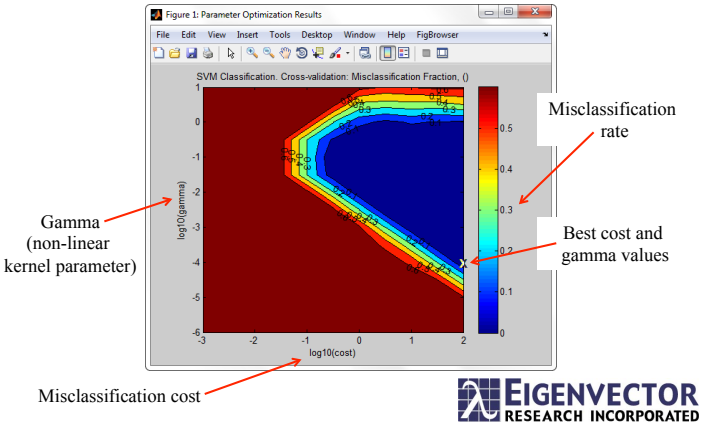
Run the model. "Optimal" parameters are found and the model built.



EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

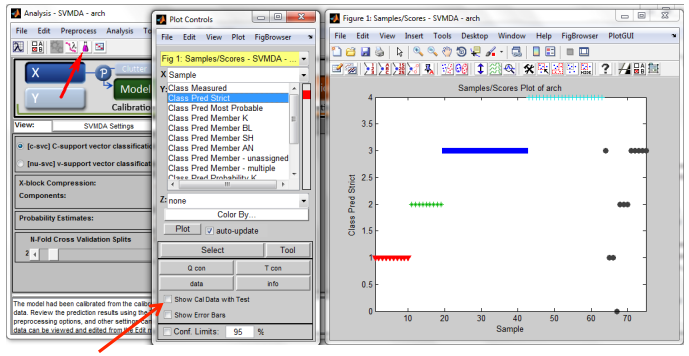
View the parameter search results. "X" identifies the optimal parameters.



EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

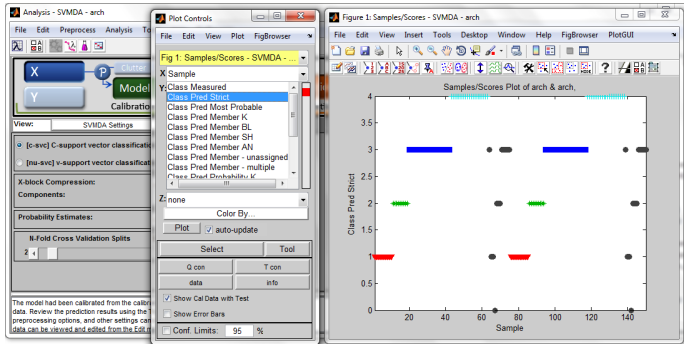
View class predictions for the training dataset



EIGENVECTOR RESEARCH INCORPORATED

SVM Classification

View class predictions for the training and test datasets



EIGENVECTOR RESEARCH INCORPORATED

Per-class probability estimates

SVMDA predicts class labels or per-class probabilities for samples. The per-class probabilities are calculated if the "Probability Estimates" option is enabled in the SVMDA analysis window (or set the option `probabilityestimates = 1` in command line usage). The method is explained in Chang and Lin (2001), section 8, "Probability Estimates".

The per-class probability estimates appear in model.detail.predprobability as an nsample x nclasses array. The columns are the classes, in the order given by model.detail.svm.model.label where the class values are what was in the input X-block.class{1} or Y-block. These probabilities are used to find the most likely class for each sample and this is saved in pred.pred{2} and model.detail.predictedclass. This is a vector of length equal to the number of samples with values equal to class values (model.detail.class{1}).

EIGENVECTOR RESEARCH INCORPORATED

An aside: confusion matrix

You can get details of the true-positive, false-positive rates, etc. by using the 'confusionmatrix' command line function.

```
>> confusionmatrix(model);

Confusion Matrix:
    Class:      TP          FP          TN          FN
        K       1.00000     0.00000     1.00000     0.00000
        BL      1.00000     0.00000     1.00000     0.00000
        SH      1.00000     0.00000     1.00000     0.00000
        AN      1.00000     0.00000     1.00000     0.00000
```

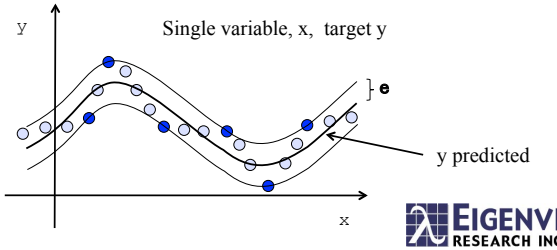
See 'help confusionmatrix' and 'help confusiontable'.

EIGENVECTOR RESEARCH INCORPORATED

SVM Regression

Goal: Predict a property of interest (y) from measured values (x)

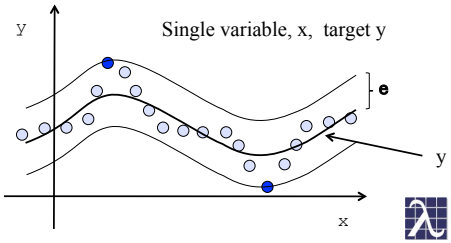
- SVM regression introduces a parameter epsilon, **e,** representing the maximum penalty-free deviation of training set predictions from the target values.
- The C parameter controls the penalty for deviations greater than **e**.



EIGENVECTOR RESEARCH INCORPORATED

SVM Regression

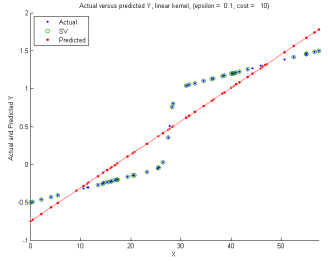
- Increasing the epsilon value allows a more relaxed fitting of the regression to the training data.
- Reduces risk of overfitting the training data but might miss important features of the data.



EIGENVECTOR RESEARCH INCORPORATED

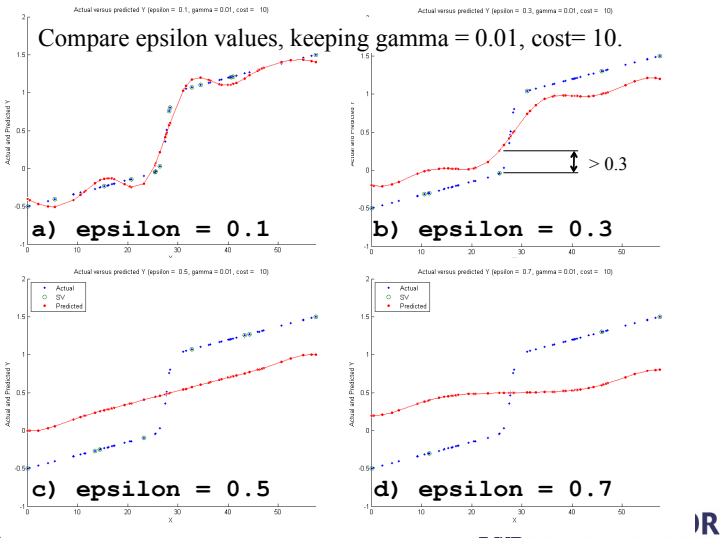
SVM Regression

Linear kernel SVM regression produces a straight line response of y with respect to x. It cannot capture any non-linearity in the x-y relation.



Regression with **linear kernel**. epsilon = 0.1, cost = 10.

EIGENVECTOR RESEARCH INCORPORATED



SVM Regression

Summary of parameter effects:

- Decreasing **epsilon** causes tighter fitting of the regression represented in the training data.
- Decreasing **cost** causes looser fitting to the training data regression relation.
- **Gamma** determines how strongly non-linear the modeled regression can be.

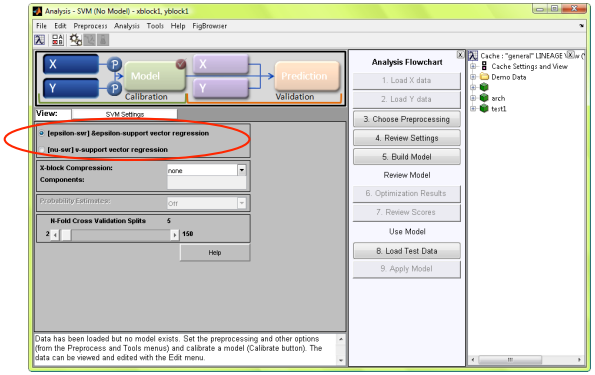
  Smaller gamma tends towards linear kernel behavior, giving a straight predicted line. Larger gamma allows the SVM to represent stronger nonlinearity in the x,y regression.

SVM Regression

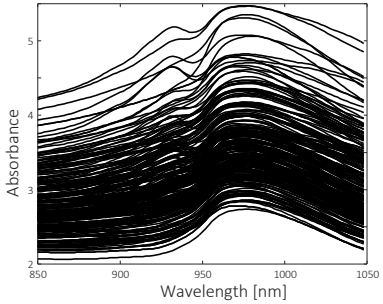
Selecting the parameters to use for SVM regression.

- SVM regression has one more parameter than SVM classification, (epsilon, cost, [gamma]). Pick optimal parameter set by scanning over parameter ranges testing for the best cross-validation RMSE. This can be slow…

SVM e-regression using PLS_Toolbox GUI



Tecator NIR calibration example



215 finely chopped pure meat samples measured by Infratec Food and Feed Analyzer.

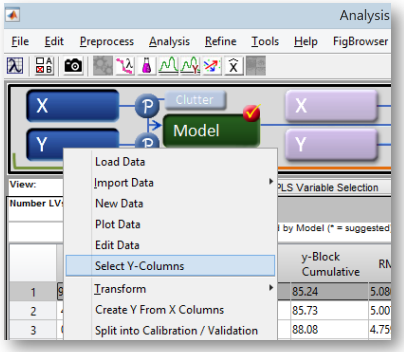
Nonlinearity is exhibited between the spectra and the fat and moisture content. The protein content only demonstrates weak nonlinearity.

Borggaard, Thodberg, Analytical Chemistry, 64 (1992) 545–551.
Thodberg, IEEE Transactions on Neural Networks 7 (1996) 56–72.
http://lib.stat.cmu.edu/datasets/tecator

EIGENVECTOR RESEARCH INCORPORATED

Tecator NIR calibration example

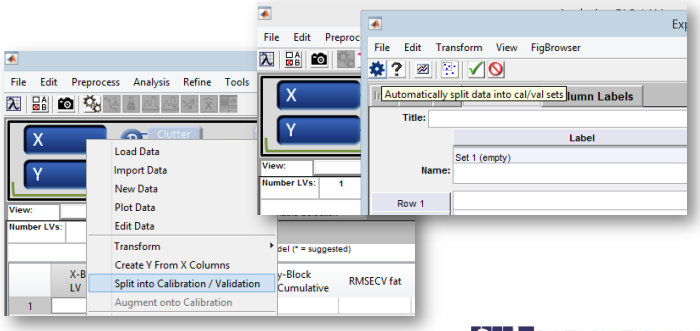
Compare how PLS and SVR predict fat



EIGENVECTOR RESEARCH INCORPORATED

Tecator NIR calibration example

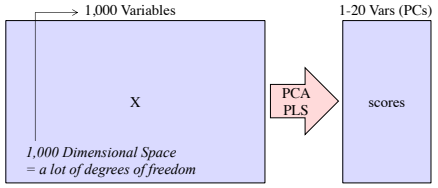
Possibly retain a validation set
Maybe only keep 20% in calibration to speed up



EIGENVECTOR RESEARCH INCORPORATED

SVM Performance Issues

- **X-block compression:** Data compression performed on x-block prior to calculating or applying the SVM model.
- 'pca' uses a simple PCA model to compress the information. 'pls' uses either a pls or plsda model (depending on the svmtype). Compression can make the SVM more stable and less prone to overfitting, and faster to calculate.



EIGENVECTOR RESEARCH INCORPORATED

SVM Performance Issues

SVM optimal parameter search: the slowest step when building an SVM model. SVM regression with Gaussian RBF involves three parameters (epsilon, cost, gamma), so searching can be slow. Linear kernel function uses one SVM parameter less than the Gaussian RBF so the search is [notably] faster.

Due to randomness in the selection of training samples during the cross-validation process, multiple runs of the optimal parameter search may return slightly different optimal parameter values.

A time limit (option 'cvtimelimit') exists for model building during CV parameter searching (default 2 sec), because searching can be exceptionally slow for some parameter combinations. If an exact parameter set is used then no CV search occurs but even then, a timelimit = 30 x cvtimelimit is enforced when building the SVM.

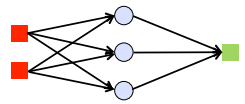
SVM References

- Ivanciuc, O. (2007). "Applications of Support Vector Machines in Chemistry" http://www.ivanciuc.org/Files/Reprint/Ivanciuc_SVM_CCR_2007_23_291.pdf
- Chang, C.-C. and C.-J. Lin, (2001) "Libsvm: a library for support vector machines". http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
- Hsu, C.W., C.C. Chang, C.J. Lin (2009). "A Practical Guide to Support Vector Classification". http://www.csie.ntu.edu.tw/~cjlin
- Smola, A.J. and B. Scholkopf (2003). "A tutorial on support vector regression". http://alex.smola.org/papers/2003/SmoSch03b.pdf
- Bennett, K.P. and C. Campbell, "Support Vector Machines: Hype or Hallelujah?" (2000). http://www.sigkdd.org/explorations/issue2-2/bennett.pdf

Artificial Neural Networks

Artificial Neural Networks

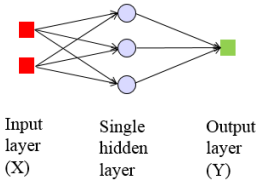
- Artificial Neural Network (ANN) is a non-linear regression method.
- ANN mimics the architecture of the brain where a network of neurons are connected by synapses. X data are presented to the ANN in the input layer. A simple single hidden-layer example:



Input layer (X) | Single hidden layer | Output layer (Y)

If the input to a neuron is strong enough the neuron is activated and it affects downstream connected neurons

ANN



ANN is defined by:

•The layers and nodes in each layer and their connections.

Input layer has as many nodes as X has variables.

Output layer has as many nodes as Y has variables.

•Weights: weight associated with each synapse, or node-pair.

•Activation function converts node's weighted input to its output, and is usually step-like such as tanh.

EIGENVECTOR RESEARCH INCORPORATED

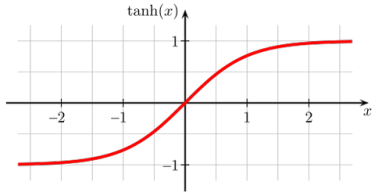
Artificial Neural Networks

Each node receives input x_i, from N upstream nodes, each modified by a transmission factor w_i:

$I = \sum_{i=1}^{N} (w_i x_i + b)$

and outputs a signal, $x = f(I)$, to downstream nodes.

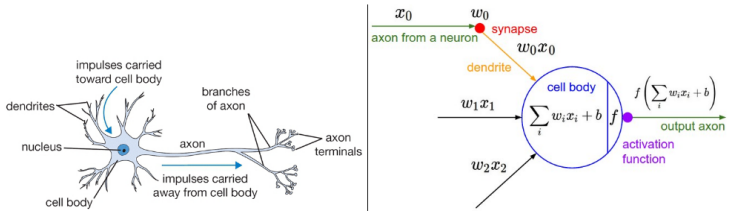
$f()$ is called the Activation Function. It non-linearly converts node's weighted input to its output, and is usually step-like such as tanh.



The activation function must be smooth (differentiable) to allow the backpropagation error reduction method to work in training the ANN

EIGENVECTOR RESEARCH INCORPORATED

Biological Analogy



A cartoon drawing of a biological neuron (left) and its mathematical model (right).

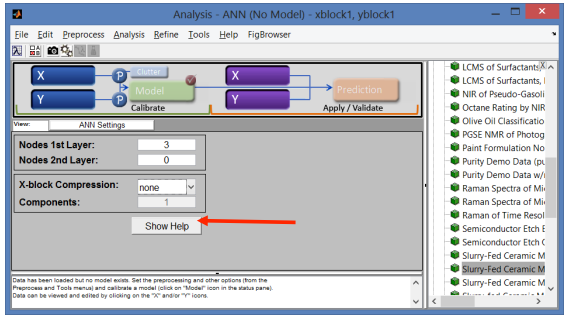
You have ~86 billion neurons and $10^{14} - 10^{15}$ synapses.

(http://cs231n.github.io/neural-networks-1/#bio)

ANNs in 2015 as large as approx. 10^{11} weights (synapses)

EIGENVECTOR RESEARCH INCORPORATED

ANN using the Analysis GUI



Training process for updating the weights for each connection. ANN uses a feedforward network with back-propagation training. The training calculates the error between actual and predicted output and adjusts weights to minimize this error.

EIGENVECTOR RESEARCH INCORPORATED

Compare using plsdata dataset

	RMSEC	RMSEP
PLS	0.1063	0.1385
LWR	0.0964	0.1765
SVM	0.0949	0.1495
ANN	0.0985	0.1449

- Both LWR, SVM and ANN have more freedom to fit data
- Data relationship is NOT inherently non-linear so freedom is overkill

- Mean Centering preprocessing the same for X in all
- PLS: 3 LVs
- LWR: 30 neighbors, 3 LVs (Global PCR)
- SVM: Radial Basis Function, no compression
- ANN: 1 hidden layer, 3 nodes, BPN method.

EIGENVECTOR RESEARCH INCORPORATED

Using NL_tank_data dataset

	RMSEC	RMSEP
PLS	0.2619	1.8937
LWR	0.1152	0.1589
SVM	0.1419	0.3654
ANN	0.1424	0.4934

- Both LWR, SVM and ANN have more freedom to fit data
- Data relationship IS non-linear so freedom is needed. PLS predicts poorly.

- Autoscale preprocessing the same for X in all
- PLS: 2 LVs
- LWR: 30 neighbors, 3 LVs (Global PCR)
- SVM: Radial Basis Function, no compression
- ANN: 1 hidden layer, 2 nodes, BPN method.

EIGENVECTOR RESEARCH INCORPORATED

Using Tecator dataset

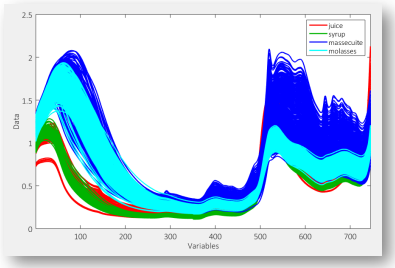
	RMSEC	RMSEP
PLS	3.891	3.743
LWR	1.138	3.326
SVM	1.252	1.262
ANN	0.605	0.825

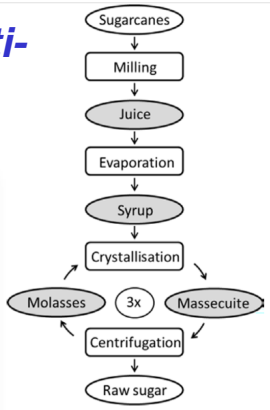
- Data relationship IS non-linear so freedom is needed. PLS predicts poorly.

- Using 50/50 split into Calibration and Validation datasets.
- SavGol derivative and mean center preprocessing the same for X in all
- PLS: 2 LVs
- LWR: 11 neighbors, 3 LVs (Global PCR)
- SVM: Radial Basis Function, no compression
- ANN: 1 hidden layer, 2 nodes, BPN method.

EIGENVECTOR RESEARCH INCORPORATED

Final example multi-step process

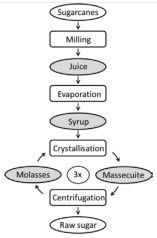




R. Tange, M. A. Rasmussen, Eizo Taira, and R. Bro. Application of support vector regression for simultaneously modelling of near infrared spectra from multiple process steps. *Journal of Near Infrared Spectroscopy 23:75-84, 2015.*

EIGENVECTOR RESEARCH INCORPORATED

Final example multi-step process

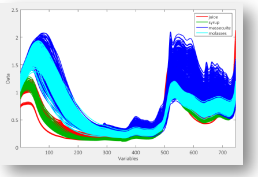


Predict Pol (sugar)

Retain a validation set keeping 30% in calibration

Compare SVM, ANN and PLS

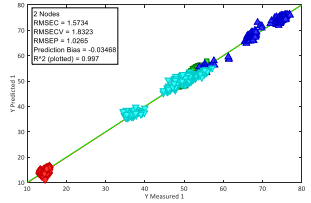
Use SNV and PCA compression



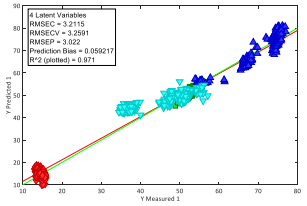
EIGENVECTOR RESEARCH INCORPORATED

Final example

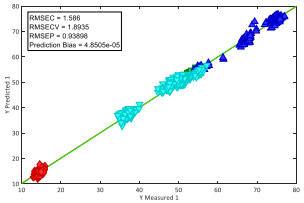
ANN



PLS



SVM



EIGENVECTOR RESEARCH INCORPORATED

Conclusions

- Many tools and approaches readily available
- Tip #1: Use your background knowledge. The more you know, the better you can do
- Prefer simpler models in general. Simpler implies less overfit
- Remember validation
- Nonlinear transformations of data together with e.g. PLS are nice as they allow easy visualization, outlier detection, variable selection etc.

EIGENVECTOR RESEARCH INCORPORATED