Granular modelling through regression analysis

Prof. A.Bargiela

Why regression model built on granulated data?

- Essence of granulation is capturing some general properties of data that are semantically distinct from the actual data values.
- Essence of regression is to find a functional description of general properties of data (as distinct from reproducing the values of dependent variables)
- → use of data that is semantically closer to the regression model

Where granulated data comes from?

- Finitely accurate measurements (reflection of approximate nature of discretisation of real-world entities) – raw data granules.
- Representation of detailed data by prototypes that capture some essential characteristics of data – processed data granules.

Plan

- Classical regression analysis
- Simple fuzzy linear regression
- Gradient descent optimisation for simple fuzzy regression
- Multiple fuzzy linear regression
- Gradient descent optimisation for multiple fuzzy regression

Classical regression

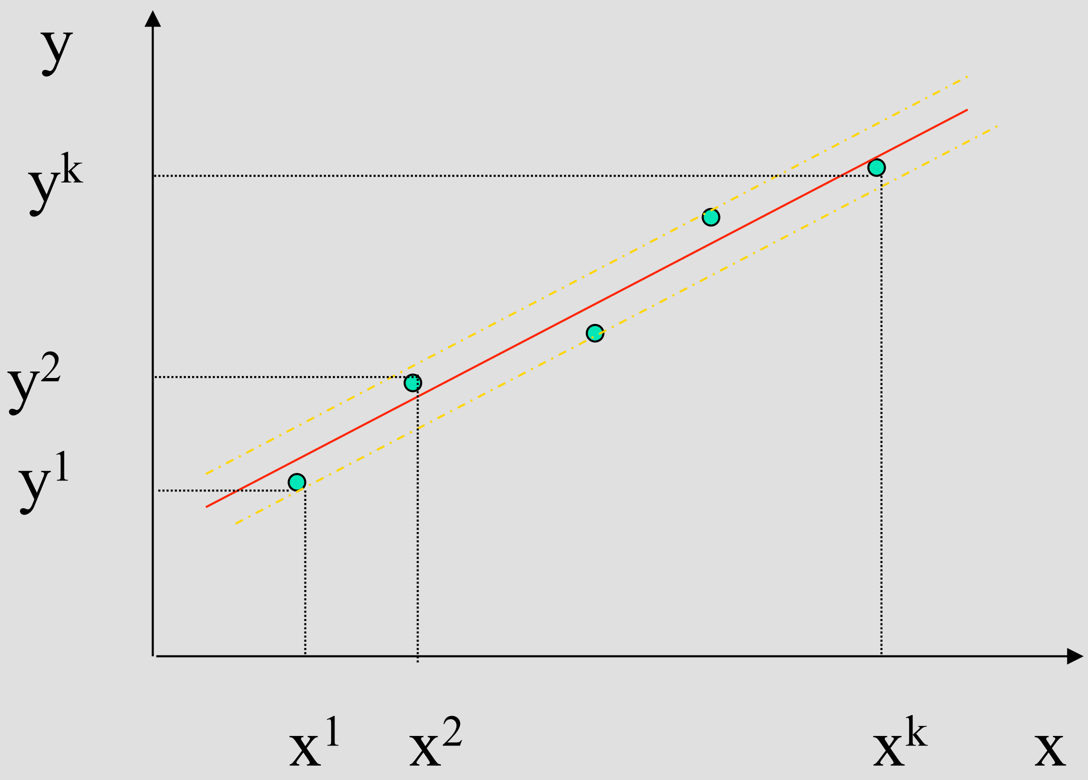
$$y = a_0 + a_1 x + \varepsilon$$

$$C(a_{0,}a_1) = \sum_{i=1}^{k} (y^i - (a_0 + a_1 x^i))^2$$

$$\frac{\partial C}{\partial a_0} = 0 \qquad \frac{\partial C}{\partial a_1} = 0$$

$$\hat{a}_0 = \frac{1}{k}\sum_{i=1}^{k} y^i - \hat{a}_1 \frac{1}{k}\sum_{i=1}^{k} x^i \qquad \hat{a}_1 = \frac{\sum_{i=1}^{k}(y^i - \frac{1}{k}\sum_{i=1}^{k} y^i)(x^i - \frac{1}{k}\sum_{i=1}^{k} x^i)}{\sum_{i=1}^{k}(x^i - \frac{1}{k}\sum_{i=1}^{k} x^i)^2}$$

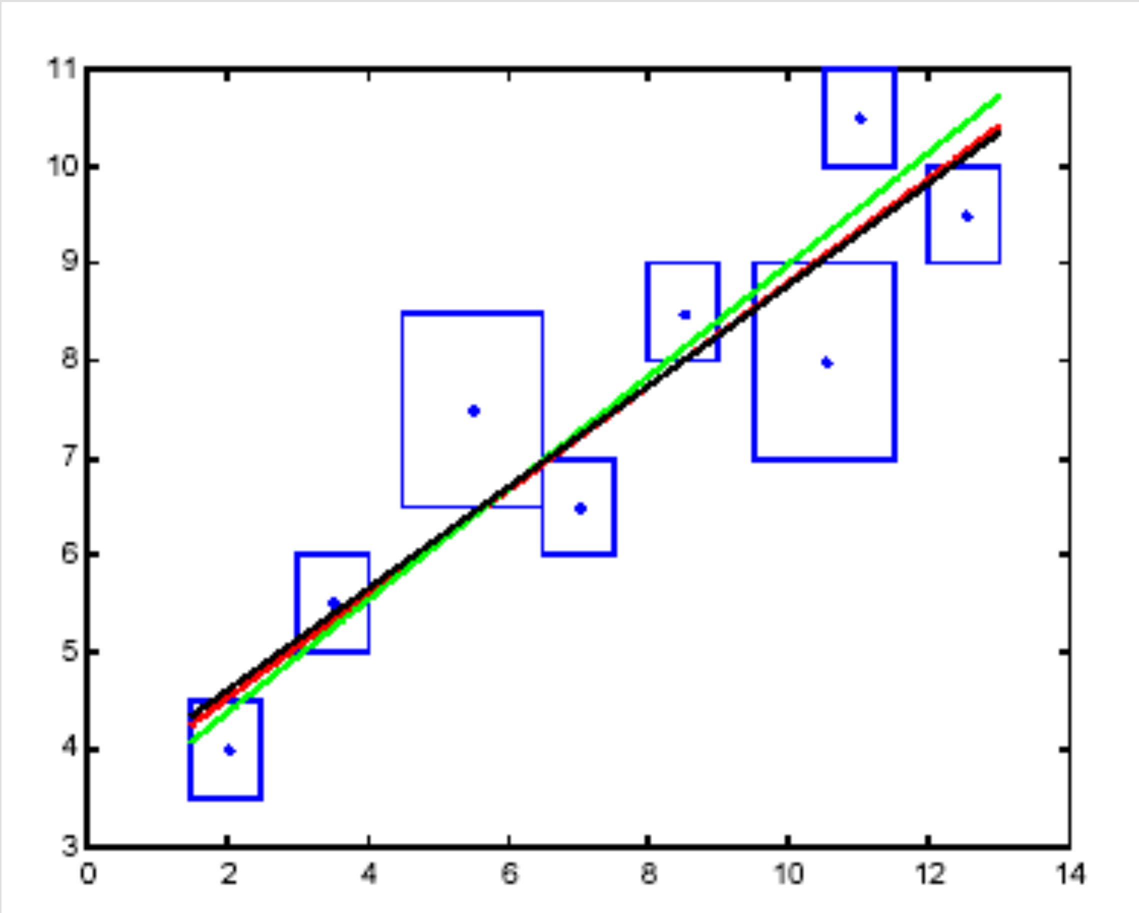
Classical regression



$$y = f(x) + \varepsilon$$

$$y = a_0 + a_1 x + \varepsilon$$

Fuzzy regression data (raw)



Fuzzy regression data (processed)

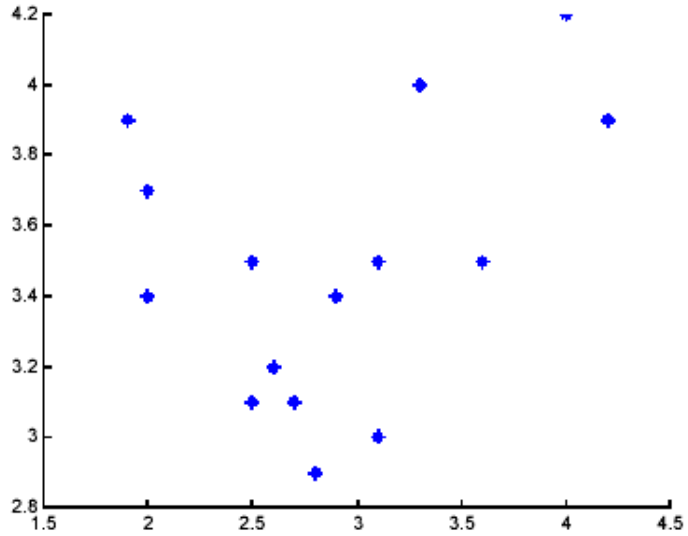
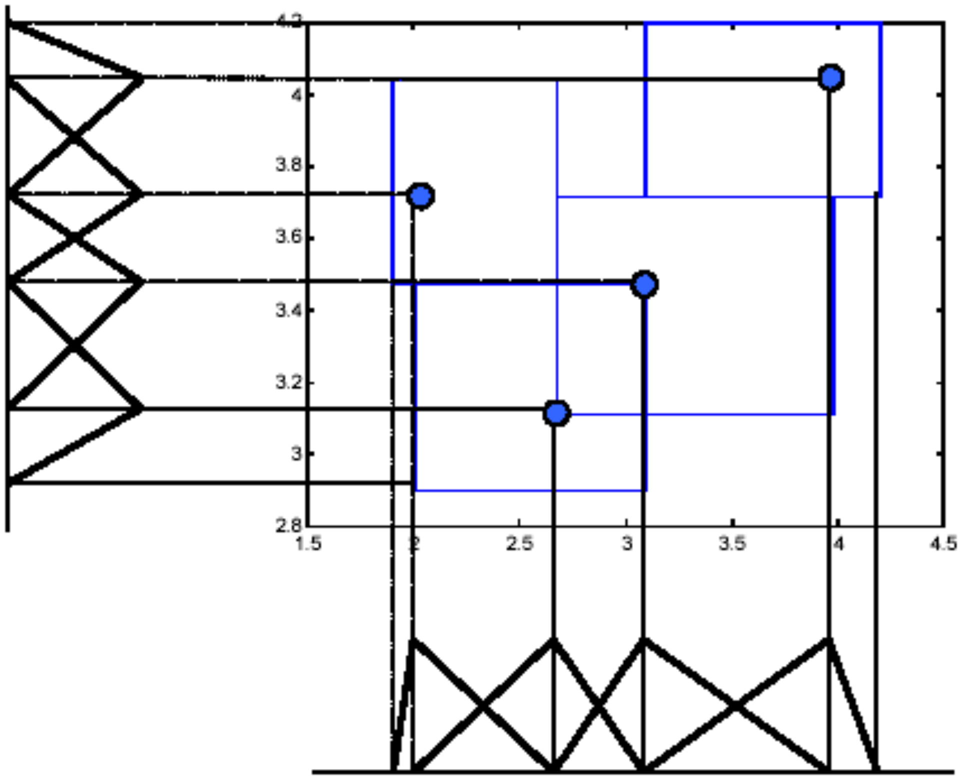
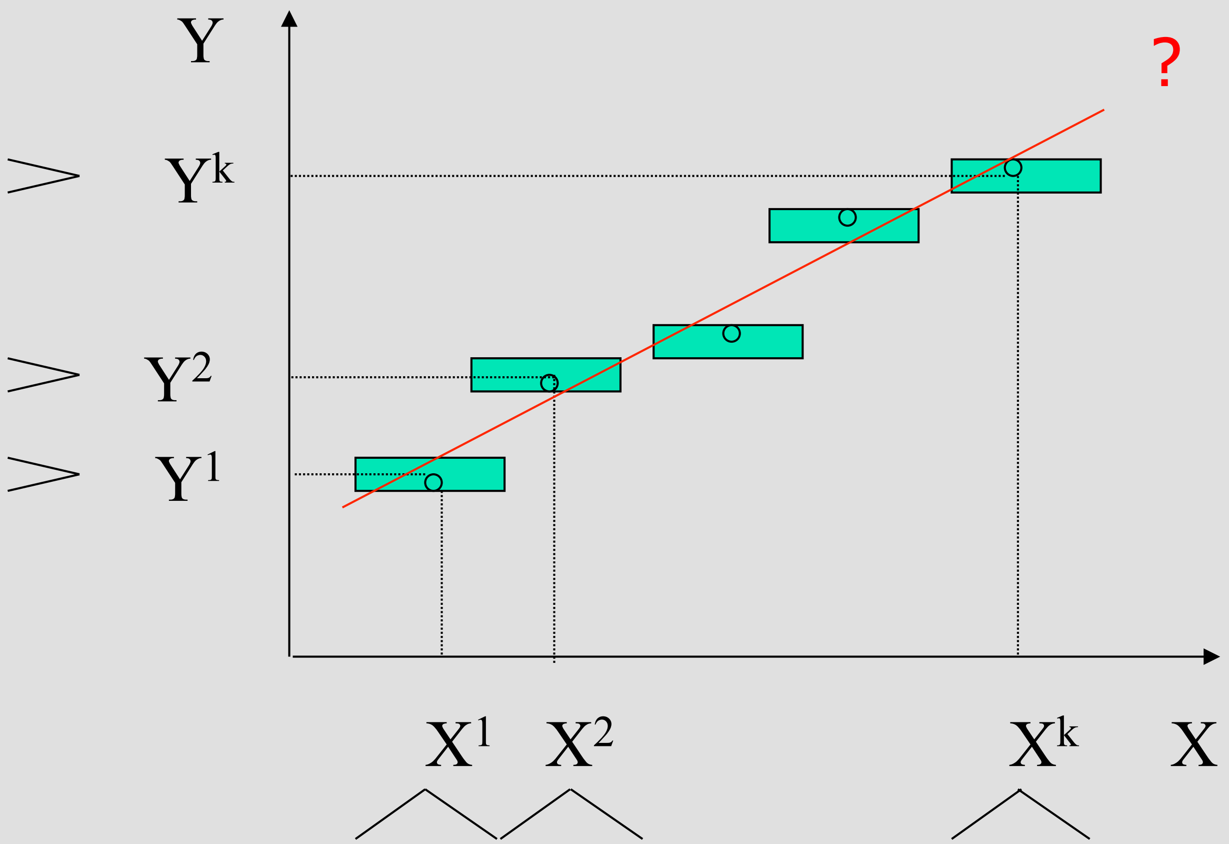


Figure 1. Original numerical data



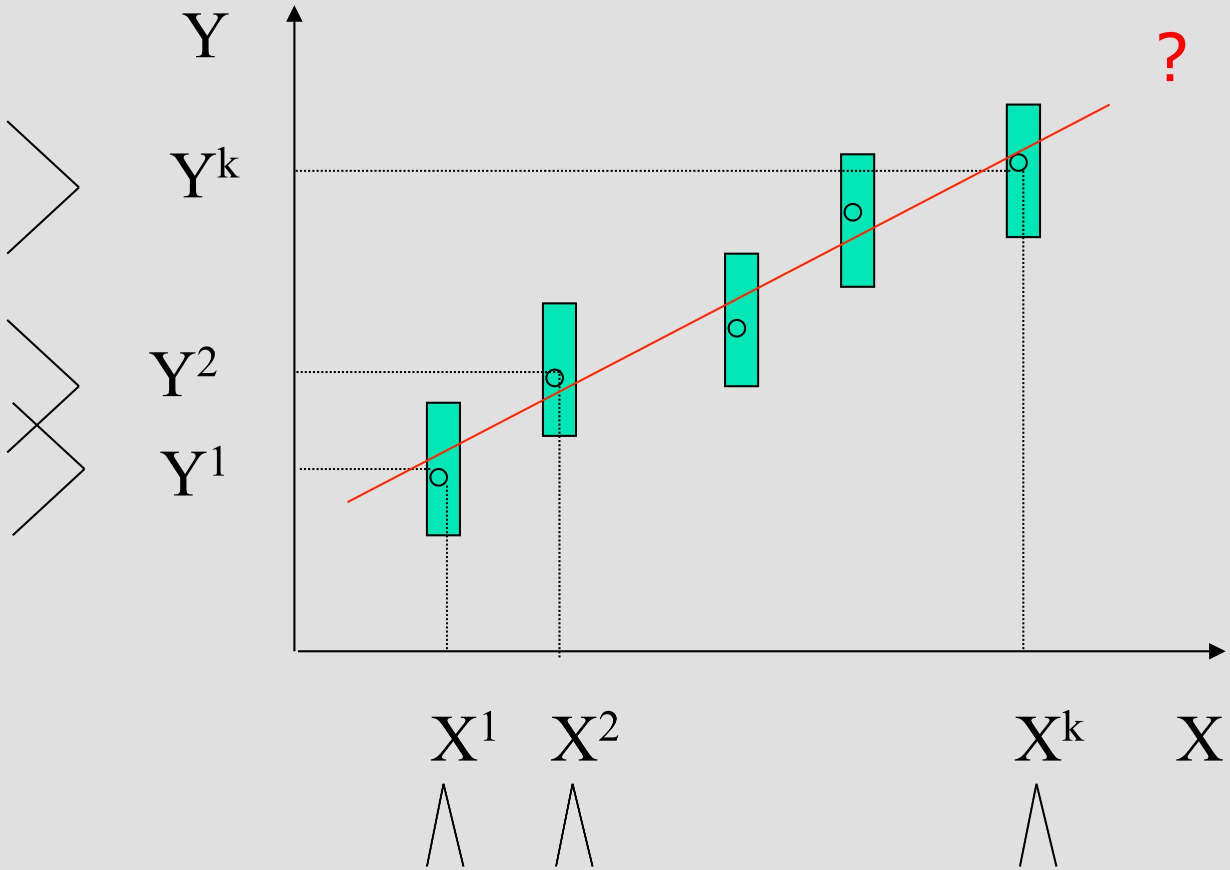
Simple fuzzy linear regression



$$Y=f(X)+\varepsilon$$

$$Y=b_0+b_1X+\varepsilon$$

Simple fuzzy linear regression

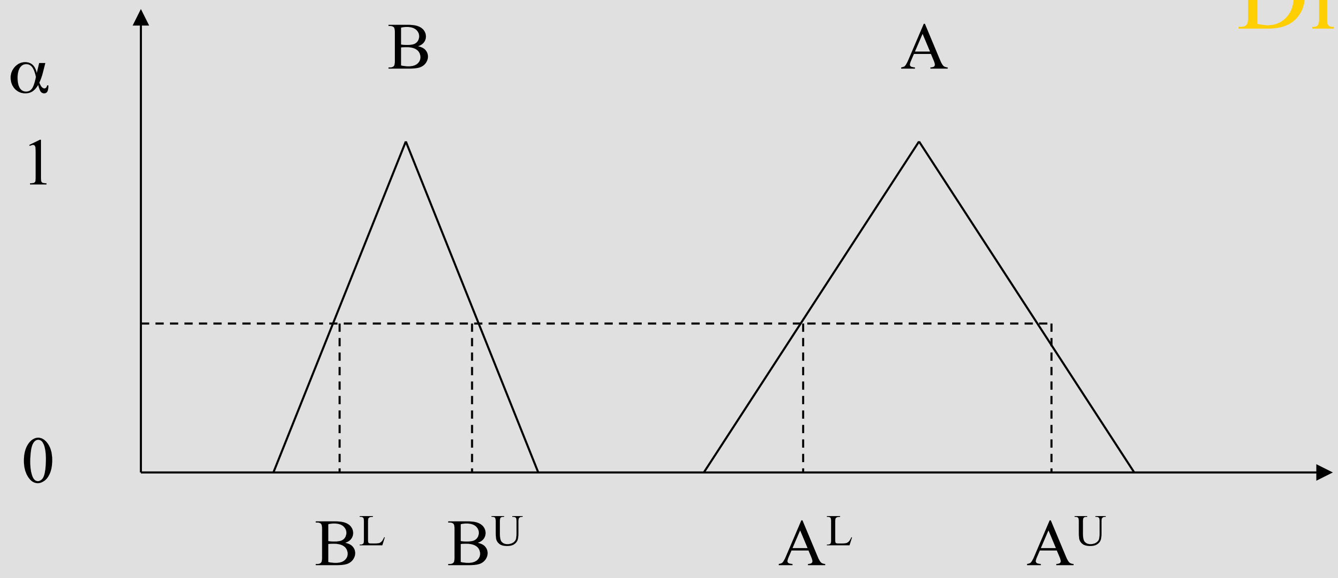


$$Y = f(X) + \varepsilon$$

$$Y = b_0 + b_1 X + \varepsilon$$

Simple fuzzy linear regression

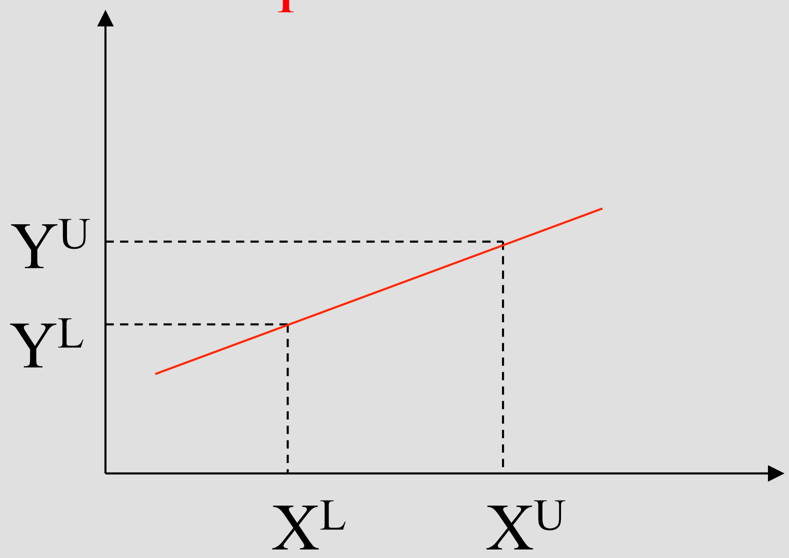
Distance



$$d(A,B)=\sqrt{\int_0^1 (A^L(\alpha)-B^L(\alpha))^2\,d\alpha+\int_0^1 (A^U(\alpha)-B^U(\alpha))^2\,d\alpha}$$

Simple fuzzy linear regression

$b_1>0$

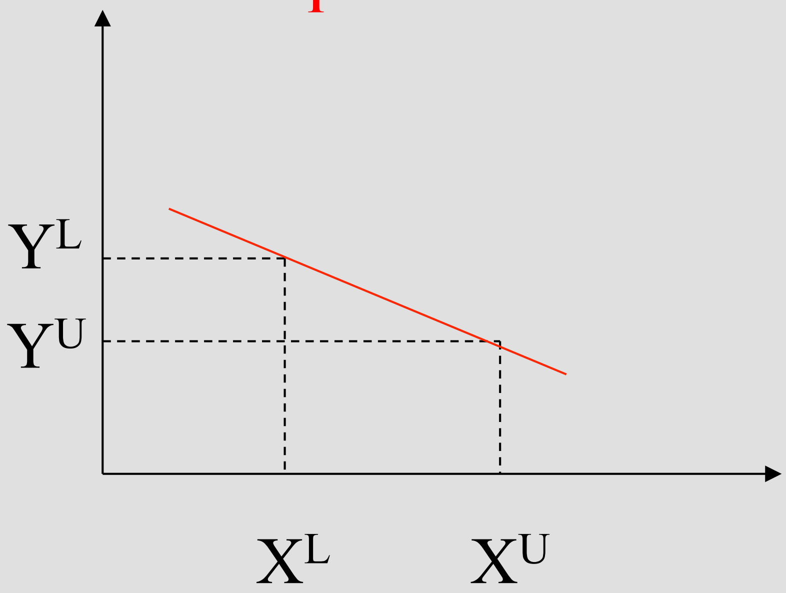


$b_1>0$

$$H^+(b_0,b_1)=\sum_{i=1}^{k}\int_0^1 (Y_i^L(\alpha)-b_0-b_1X_i^L(\alpha))^2\,d\alpha + \sum_{i=1}^{k}\int_0^1 (Y_i^U(\alpha)-b_0-b_1X_i^U(\alpha))^2\,d\alpha$$

Simple fuzzy linear regression

$b_1<0$



$b_1<0$

$$H^{-}(b_0,b_1)=\sum_{i=1}^{k}\int_0^1 (Y_i^U(\alpha)-b_0-b_1X_i^L(\alpha))^2\,d\alpha$$
$$+\sum_{i=1}^{k}\int_0^1 (Y_i^L(\alpha)-b_0-b_1X_i^U(\alpha))^2\,d\alpha$$

Simple fuzzy linear regression

$b_1 > 0$

$$\hat{b}_0^+ = \tilde{Y} - \hat{b}_1^+ \tilde{X}$$

$$\hat{b}_1^+ = \frac{SS_{xy}^+}{SS_{xx}}$$

$b_1 < 0$

$$\hat{b}_0^- = \tilde{Y} - \hat{b}_1^- \tilde{X}$$

$$\hat{b}_1^- = \frac{SS_{xy}^-}{SS_{xx}}$$

Simple fuzzy linear regression

$$\tilde{X} = \int_0^1 \frac{\overline{X}^L(\alpha) + \overline{X}^U(\alpha)}{2} d\alpha \qquad \tilde{Y} = \int_0^1 \frac{\overline{Y}^L(\alpha) + \overline{Y}^U(\alpha)}{2} d\alpha$$

$$SS_{xx} = \sum_{i=1}^{k} \int_0^1 ((X_i^L(\alpha))^2 + (X_i^U(\alpha))^2) d\alpha - 2k\tilde{X}^2$$

$b_1 > 0$ $$SS_{xy}^{+} = \sum_{i=1}^{k} \int_0^1 (X_i^L(\alpha) Y_i^L(\alpha) + X_i^U(\alpha) Y_i^U(\alpha)) d\alpha - 2k\tilde{X}\tilde{Y}$$

$b_1 < 0$ $$SS_{xy}^{-} = \sum_{i=1}^{k} \int_0^1 (X_i^U(\alpha) Y_i^L(\alpha) + X_i^L(\alpha) Y_i^U(\alpha)) d\alpha - 2k\tilde{X}\tilde{Y}$$

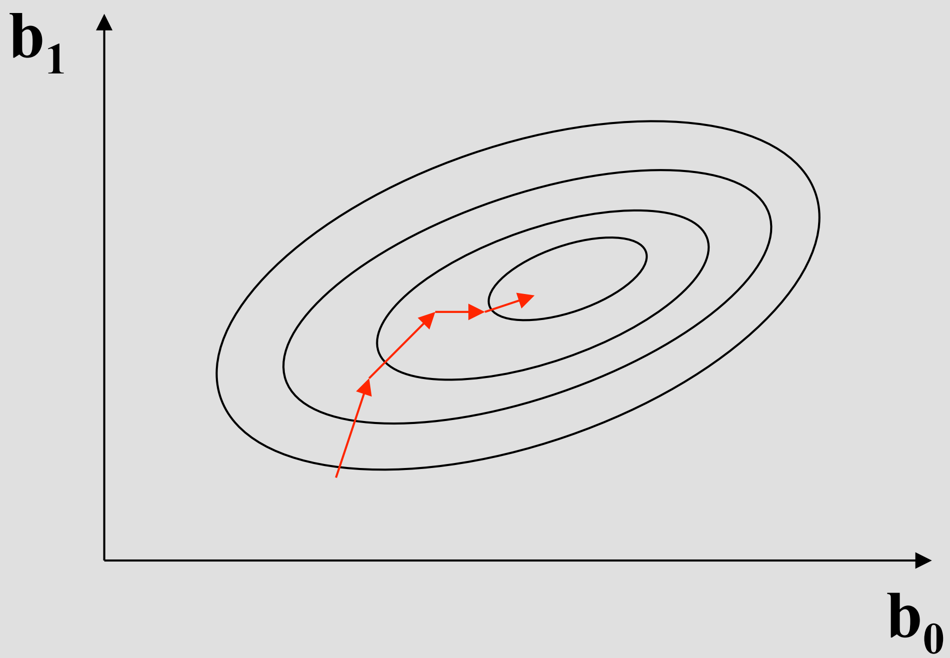
Simple fuzzy linear regression

Analytical solution for simple fuzzy linear regression does not scale-up easily to multiple linear regression because of the need to consider every permutation of positive/negative slope of the regression line between each independent and dependent variable.

Gradient descent optimisation

Idea for scaling-up: avoid analytical solution by conducting an iterative refinement of some initial guess of the parameters of the regression line

$$H(b_0, b_1)$$



Gradient descent optimisation

$$\Delta b_0 = \mu_0 \frac{\partial H^*(b_0, b_1)}{\partial b_0}$$

$$\Delta b_1 = \mu_1 \frac{\partial H^*(b_0, b_1)}{\partial b_1}$$

$$b_0^i = b_0^{i-1} + \Delta b_0$$

$$b_1^i = b_1^{i-1} + \Delta b_1$$

Gradient descent optimisation

The algorithm:

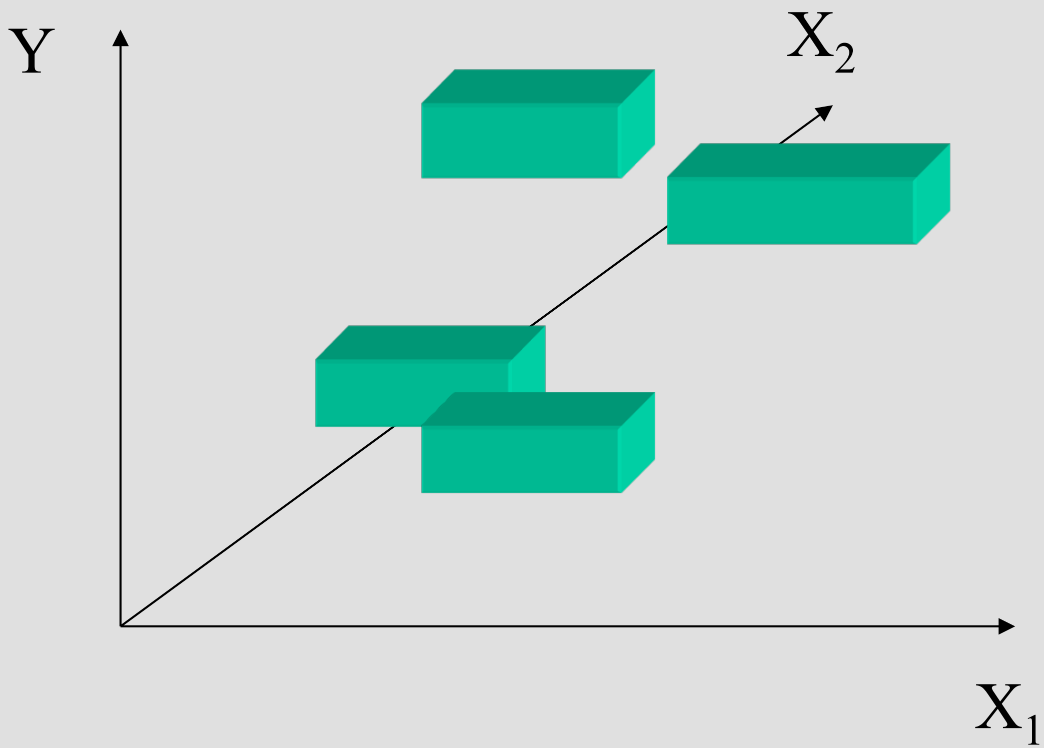
1. **Calculate gradient of the error function with respect of individual regression variables**
2. **Calculate the change to regression variables implied by the gradient and the learning coefficient**
3. **Iterate until the updates become negligible**

Gradient descent optimisation

Factors affecting convergence:

1. **Learning rate μ**
2. **Regularity of the error function (achieved by normalising regression variables)**

Multiple fuzzy linear regression



Multiple fuzzy linear regression

$$Y = b_0 + b_1 X^1 + b_2 X^2 + \ldots + b_m X^m$$

$$\min H(b_0, b_1, \ldots, b_m) =$$

$$\sum_{i=1}^{k} d^2 (Y_i, b_0 + b_1 X_i^1 + b_2 X_i^2 + \ldots + b_m X_i^m)$$

Gradient descent optimisation

$$\Delta b_0 = \mu_0 \frac{\partial \widehat{H}(b_0,\ldots,b_m)}{\partial b_0}$$

$$\Delta b_j = \mu_j \frac{\partial \widehat{H}(b_0,\ldots,b_m)}{\partial b_j} \qquad j=1,..,m$$

$$b_j^i = b_j^{i-1} - \Delta b_j \qquad j=0,..,m$$

Gradient descent optimisation

$$\frac{\partial \widehat{H}(b_0,\ldots,b_m)}{\partial b_0} = -2\widetilde{Y} + 4kb_0 + 2b_1\widetilde{\widehat{X}}^1 + \ldots + 2b_m\widetilde{\widehat{X}}^m$$

$$\frac{\partial \widehat{H}(b_0,\ldots,b_m)}{\partial b_j} = -2\overleftrightarrow{SS}_{X^jY} + 2b_0\widetilde{\widehat{X}}^j + 2b_1\overleftrightarrow{SS}_{X^jX^1} + \ldots + 2b_m\overleftrightarrow{SS}_{X^jX^m}$$

$$\widetilde{Y} = \int_0^1 \left(\sum_{i=1}^{k} Y_i^L(\alpha) + \sum_{i=1}^{k} Y_i^U(\alpha)\right)d\alpha \qquad \widetilde{\widehat{X}}^j = \int_0^1 \left(\sum_{i=1}^{k} \widehat{X}_i^{jL}(\alpha) + \sum_{i=1}^{k} \widehat{X}_i^{jU}(\alpha)\right)da$$

$$\overleftrightarrow{SS}_{X^jY} = \int_0^1 \sum_{i=1}^{k} \left(Y_i^L(\alpha)\widehat{X}_i^{jL}(\alpha) + Y_i^U(\alpha)\widehat{X}_i^{jU}(\alpha)\right)d\alpha$$

$$\overleftrightarrow{SS}_{X^jX^p} = \int_0^1 \sum_{i=1}^{k} \left(\widehat{X}_i^{pL}(\alpha)\widehat{X}_i^{jL}(\alpha) + \widehat{X}_i^{pU}(\alpha)\widehat{X}_i^{jU}(\alpha)\right)d\alpha$$

Gradient descent optimisation

The algorithm:

1. **Calculate gradient of the error function with respect of individual regression variables**
2. **Calculate the change to regression variables implied by the gradient and the learning coefficient**
3. **Iterate until the updates become negligible**

Gradient descent optimisation

Computational advantage:

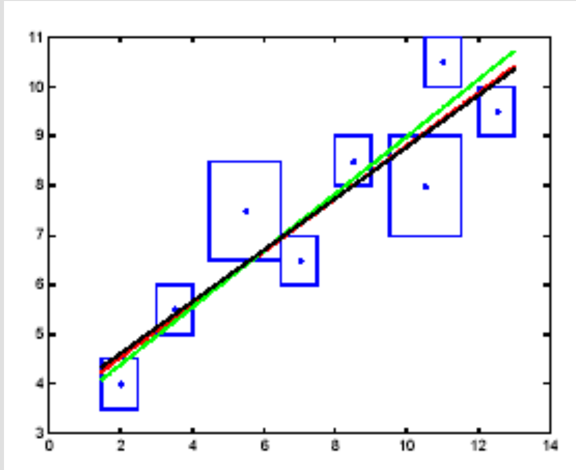
Since we know, at the beginning of each iteration, the values of estimates of b_1 we can swap-round the upper/lower limits of the alpha-cuts of the independent variable if $b_1 < 0$

This means that in each iteration we consider only one specific cost function.

Example1 (Kao & Chyu)

Independent variable	Dependent variable
(1..5, 2.0, 2.5)	(3.5, 4.0, 4.5)
(3.0, 3.5, 4.0)	(5.0, 5.5, 6.0)
(4.5, 5.5, 6.5)	(6.5, 7.5, 8.5)
(6.5, 7.0, 7.5)	(6.0, 6.5, 7.0)
(8.0, 8.5, 9.0)	(8.0, 8.5, 9.0)
(9.5, 10.5, 11.5)	(7.0, 8.0, 9.0)
(10.5, 11.0, 11.5)	(10.0, 10.5, 11.0)
(12.0, 12.5, 13.0)	(9.0, 9.5, 10.0)

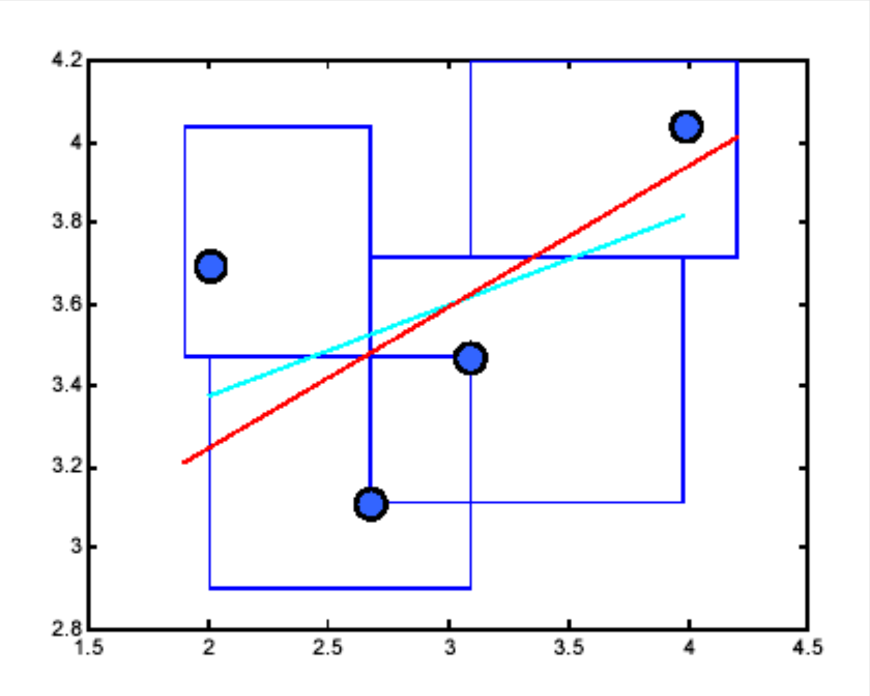
Kao C., Chyu C.L., Least-squares estimates in fuzzy regression analysis, *Eur. J. of Oper. Res.*, 148, 2, 2003, 426-435

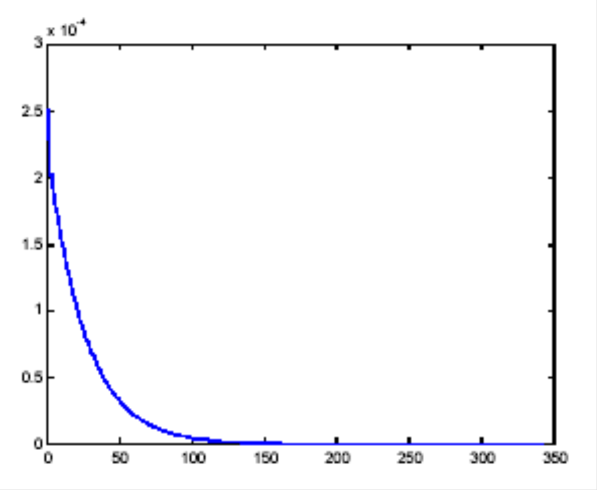


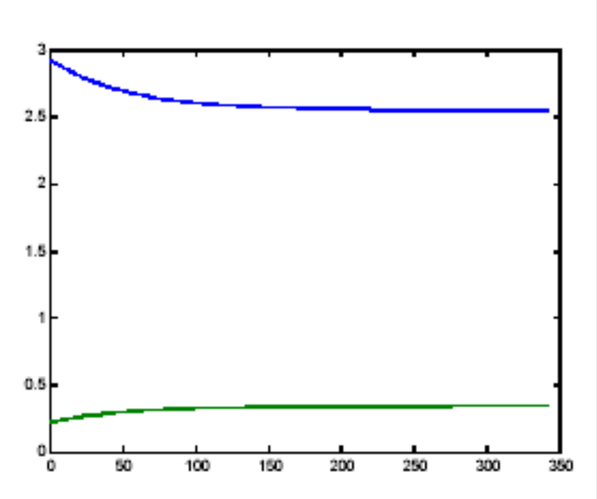
$$\mathrm{RMSE} = \sqrt{\frac{1}{c}\sum_{i=1}^{c}\left(\left|Y_i - Y_i^*\right|\right)^2}$$

Root-mean-squared error:
-Kao: **0.4864**
- proposed: **0.4862**

Example2 (granulated data)







Regression lines:
b=[2.925, 0.224] – FCM prototypes
b=[2.550, 0.348] – fuzzy variables

Example3 (Boston housing data from MLR)

http://www.ics.uci.edu/~mlearn

506 records
13 continuous attributes
1 binary-valued attribute

Data normalisation:

$$d_{norm} = \frac{d - d_{\min}}{d_{\max} - d_{\min}}$$

Regression performance on training data

prototypes	RMSE-f-base	RMSE-n-base
15	0.0703	0.0002
20	0.0245	0.0071
25	0.0315	0.0047
30	0.0239	0.0348

Regression performance on test data

prototypes	RMSE-f	RMSE-n
15	0.5091	16.8175
20	0.0993	24.3315
25	0.0578	5.7300
30	0.0837	3.4781

Conclusions

- Fuzzy multiple linear regression an important tool for generalising real-life relationships between fuzzy variables
- Computational complexity lower than that of analytical solution
- Robust convergence
- Improved generalisation ability