Granular Computing:

Granular Worlds and Applications

Index

1. Networking of Granular Worlds: Collaborative Clustering
2. Interval State Estimation in Systems Modelling
3. Information Structuring in NL Communication

Collaborative processing

Distributed sources of data (databases)

No direct interaction between databases

Goal: reveal structure common to all databases through a process of collaboration

Approach: explore possibilities of collaboration through granular communication

Horizontal collaboration

Databases about clients at various institutions

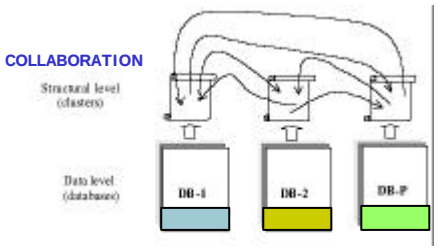
Confidentiality of data: no direct sharing of data

Sharing information granules (level of granularity)

(communication at the level of information granules)

Horizontal collaboration

Databases about same N clients at various institutions



xxxx - - - - - - - - - 1 . . N

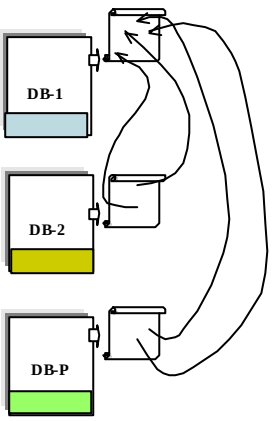
- - - - yyy - - - - - 1 . . N

- - - - - - - zzzzz 1 . . N

Confidentiality : no <u>direct</u> sharing of data

Vertical collaboration

Databases about various clients at several institutions



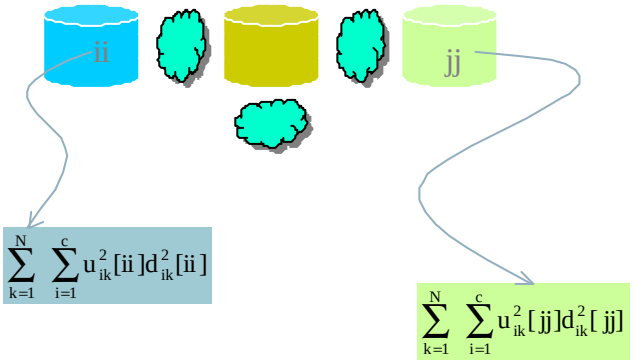
xxxx 1 . . N1

yyyy 1 . . N2

zzzz 1 . . Np

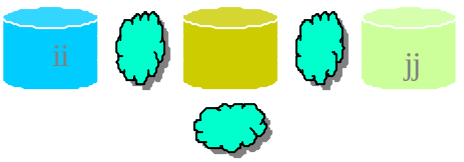
Horizontal collaboration: Fuzzy C-Means

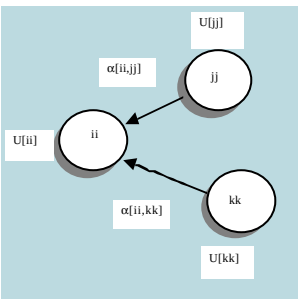
- Identify patterns by optimising local objective functions in each data set



Horizontal collaboration: Fuzzy C-Means

- Coordinate partition matrices in data sets





Communication and collaboration via partition matrices U[ii], U[jj]

Horizontal collaboration: Computational details

Modified objective function

$$Q[ii] = \sum_{k=1}^{N} \sum_{i=1}^{c} u_{ik}^2[ii] d_{ik}^2[ii] + \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} \alpha[ii, jj] \sum_{k=1}^{N} \sum_{i=1}^{c} \{u_{ik}[ii] - u_{ik}[jj]\}^2 d_{ik}^2[ii]$$

collaboration

Optimization problem

Min Q[ii]
subject to
U[ii]∈ U

$$U = =. \{u_{ik}[ii] \in [0,1] \mid \sum_{i=1}^{c} u_{ik}[ii] = 1 \text{ for all k and } 0 < \sum_{k=1}^{N} u_{ik}[ii] < N \quad \text{for i}\}.$$

Horizontal collaboration: Computational details

Unconstrained optimisation of V with respect of partition matrix and Lagrange multipliers

$$V[ii] = \sum^{N} \sum^{c} u^{2}[ii] d^{2}[ii] + \sum^{P} \alpha[ii, jj] \sum^{N} \sum^{c} \{u^{2}[ii] - u^{2}[jj]\} d^{2}[ii] - \lambda (\sum^{c} u^{2}[ii] - 1)$$

The necessary conditions

$$\frac{\partial V[ii]}{\partial u_{st}[ii]} = 0, \qquad \frac{\partial V[ii]}{\partial ë} = 0$$

Horizontal collaboration: Computational details

$$\frac{\partial V[ii]}{\partial u_{st}[ii]} = 2u_{st}[ii]d_{st}^2[ii] + 2\sum_{\substack{jj=1\\jj\neq ii}}^{p} á[ii,\ jj](u_{st}[ii] - u_{st}[jj])d_{st}^2[ii] - ë = 0$$

$$u_{st}[ii] = \frac{ë + 2d_{st}^2[ii]\left(\sum_{\substack{jj=1\\jj\neq ii}}^{p} á[ii,\ jj]u_{st}[jj]\right)}{2d_{st}^2[ii]\left(1 + \sum_{\substack{jj=1\\jj\neq ii}}^{p} á[ii,\ jj]u_{st}[jj]\right)}$$

Horizontal collaboration: Computational details

$$j_{st}[ii] = \sum_{\substack{jj=1\\jj\neq ii}}^{P} á[ii,\ jj]u_{st}[jj]$$

$$ø[ii] = \sum_{\substack{jj=1\\jj\neq ii}}^{P} á[ii,\ jj]$$

$$\sum_{s=1}^{c} u_{st}[ii] = 1$$

$$ë = \frac{1 - \sum_{s=1}^{c} \frac{j_{st}[ii]}{1 + ø[ii]}}{\sum_{s=1}^{c} \frac{1}{2d_{st}^2[ii](1 + ø[ii])}}$$

Horizontal collaboration: Partition matrix

$$u_{st}[ii] = \frac{\varphi_{st}[ii]}{1+\psi[ii]} + \frac{1}{\sum_{j=1}^{c} \frac{d_{st}^2}{d_{jt}^2}} \left[1 - \sum_{j=1}^{c} \frac{\varphi_{jt}[ii]}{1+\psi[ii]}\right]$$

Horizontal collaboration: Prototypes

$$v_s[ii] = \frac{A_s[ii] + C_s[ii]}{B_s[ii] + D_s[ii]}$$

$$A = \sum_{k=1}^{N[ii]} u_{sk}^2[ii] x_{kt}[ii]$$

$$B = \sum_{k=1}^{N[ii]} u_{sk}^2[ii]$$

$$C = \sum_{\substack{jj=1 \\ jj \neq ii}}^{p} á[ii, jj] \sum_{k=1}^{N[ii]} (u_{sk}[ii] - u_{\tilde{s}k}[ii][jj])^2 x_{kt}[ii]$$

$$D = \sum_{\substack{jj=1 \\ jj \neq ii}}^{p} á[ii, jj] \sum_{k=1}^{N[ii]} (u_{sk}[ii] - u_{\tilde{s}k}[ii][jj])^2$$

Example: Boston housing data

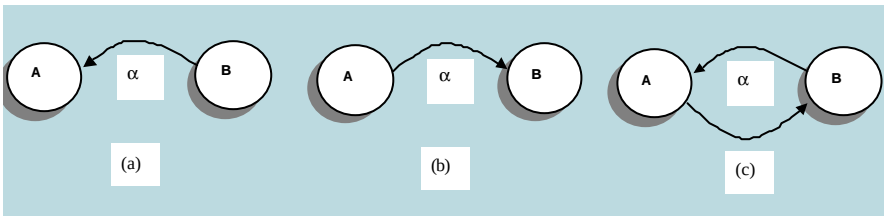
14 features split into two groups (databases) A (ii=1) and B (ii=2)

A ={per capita crime rate by town, nitric oxides concentration (parts per 10 million), proportion of owner-occupied units built prior to 1940, weighted distances to five Boston employment centers, pupil-teacher ratio by town, % lower status of the population, median value of owner-occupied homes in $1000's}

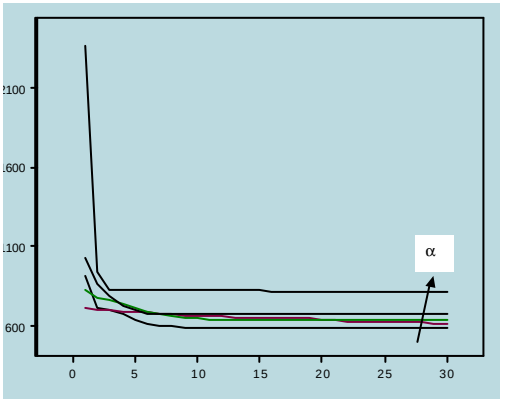
B ={ proportion of residential land zoned for lots over 25,000 sq.ft, proportion of non-retail business acres per town, Charles River dummy variable (equal to 1 if tract bounds river; 0 otherwise), average number of rooms per dwelling, index of accessibility to radial highways, full-value property-tax rate per $10,000, 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town}

Example: Boston housing data

Scenarios of collaboration:



Boston housing data



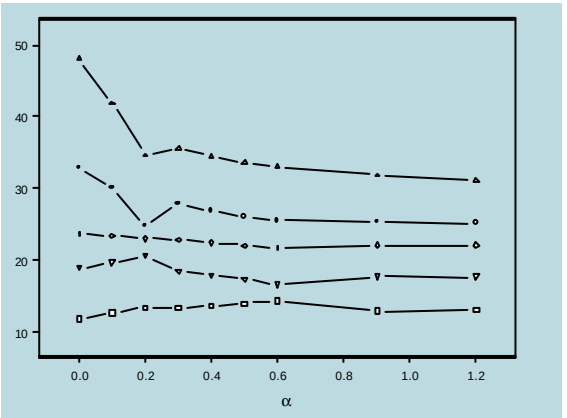
Quantification of collaboration

- Changes of prototypes
- Changes of membership grades (vis-à-vis situations with no collaboration)

Quantification of collaboration(1)

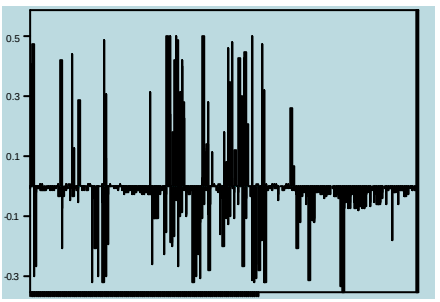
•Changes of prototypes

price of real estate

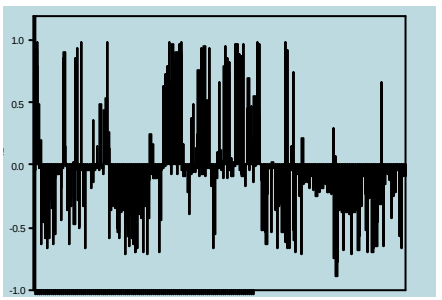


Quantification of collaboration (2)

•Changes of membership grades



α=0.2



α=0.5

Interval State Estimation in Systems Modelling

State Estimation: definition

Identify a *state uncertainty set* X(.)

$$X(M, \mathbf{z}^-, \mathbf{z}^+) := \{ \mathbf{x} \in \mathbf{R}^n : g(\mathbf{x}) \in Z(M, \mathbf{z}^-, \mathbf{z}^+) \}$$

where Z(.) is a *feasible measurements set*

$$Z(M, \mathbf{z}^-, \mathbf{z}^+) := \{ \mathbf{z}^o \in \mathbf{R}^m : z_i^- \leq z_i \leq z_i^+ , i=1,\ldots,m \}$$

State Estimation: definition

Using interval notation the *measurement set is*

$$Z(M, \mathbf{z}^-, \mathbf{z}^+) := [\mathbf{z}]$$

and the *state uncertainty set* is

$$X([\mathbf{z}])$$

State Estimation: definition

The enclosure of the state uncertainty set is defined by the uncertainty bounds on system state variables

$$[\mathbf{x}]=X^*([\mathbf{z}]) := \{ \mathbf{x} \in \mathbf{R}^n : x_i^- \leq x_i \leq x_i^+ , i=1,\ldots,n \}$$

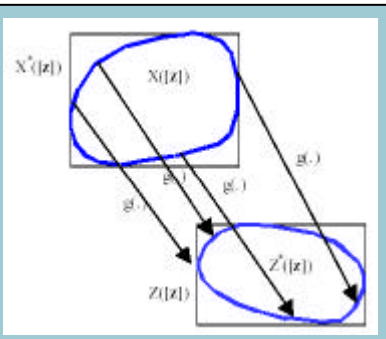
where

$$x_i^- := \min_{\mathbf{x} \in X([\mathbf{z}])} x_i, \quad i=1,\ldots,n$$

$$x_i^+ := \max_{\mathbf{x} \in X([\mathbf{z}])} x_i, \quad i=1,\ldots,n$$

State Estimation: definition

Relationship between sets



X([z]) state uncertainty set

X*([z]) enclosure of X(.)

Z([z]) measurement uncertainty set

Z*([z]) image of X(.) through g(.)

Monte Carlo method

The reference state estimates

The Monte Carlo confidence limit algorithm

1. Select a large number, k (to limit the number of iterations) and set $i = 0$.
2. Set $i = i + 1$.
3. Select a sequence of m random numbers, $r_1^i, \ldots, r_m^i$, and use these to construct a random measurement vector $\mathbf{z}^i$ from $Z([\mathbf{z}])$ as indicated in (7).
4. Calculate a state estimate, $\mathbf{x}^i$ from $\mathbf{z}^i$. If $g(\mathbf{x}^i) \in Z([\mathbf{z}])$, then use $\mathbf{x}^i$ tu update $\mathbf{x}^{iI}$, $\mathbf{x}^{iu}$. Otherwise reject $\mathbf{x}^i$ as infeasible.
5. If $i < k$, go back to step 2. Otherwise stop.

$$X^j := \{ \mathbf{x}^j \in \mathbf{R}^n : g(\mathbf{x}^i) \in Z([\mathbf{z}]) \text{ for some } i \in \{1,\ldots,j\}\}, j=1,\ldots,k$$

$$X^j \subseteq X^k \subseteq X([\mathbf{z}])$$

Linear Programming method

The application of LP requires linearisation of the state uncertainty set X([z])

Using 1-st order expansion of g(x)

$$g(\mathbf{x}) \approx g(\hat{\mathbf{x}}) + J.(\mathbf{x} - \hat{\mathbf{x}})$$

The linearised state uncertainty set is

$$X^1([\mathbf{z}]) := \{ \mathbf{x} \in \mathbf{R}^n : g(\hat{\mathbf{x}}) + J.(\mathbf{x} - \hat{\mathbf{x}}) \in Z([\mathbf{z}]) \}$$

$$X^1([\mathbf{z}]) := \{ \mathbf{x} \in \mathbf{R}^n : \mathbf{x} = \hat{\mathbf{x}} + \mathbf{dx},\ \mathbf{z}^- - g(\hat{\mathbf{x}}) \leq J.\mathbf{dx} \leq \mathbf{z}^+ - g(\hat{\mathbf{x}}) \}$$

Linear Programming method

The linearised state uncertainty set has, in general, complex topology so that an ENCLOSURE of the set is sought as

$$x^1_i{}^- := \min_{\mathbf{x} \in X^1([\mathbf{z}])} x_i, \qquad i=1,\ldots,n$$

$$x^1_i{}^+ := \max_{\mathbf{x} \in X^1([\mathbf{z}])} x_i, \qquad i=1,\ldots,n$$

Linear Programming method

The task can be expressed as 2n linear programs

$$\text{minimize } dx_i$$
$$\text{subject to } \mathbf{dz}^- \leq J.\mathbf{dx} \leq \mathbf{dz}^+$$

$$\text{maximize } dx_i$$
$$\text{subject to } \mathbf{dz}^- \leq J.\mathbf{dx} \leq \mathbf{dz}^+$$

Linear Programming method

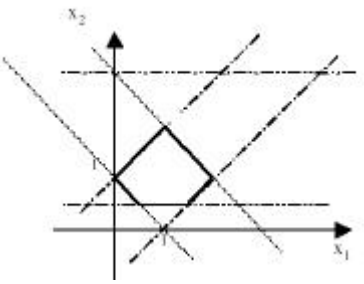
The linear programming confidence limit algorithm

1. Select an observable subset of M containing n measurements. This is the minimal measurement set and is denoted by M'. Order M with the elements of M' appearing first.
2. Re-order $\mathbf{dz}^-$ and $\mathbf{dz}^+$ according to the new ordering of M. Assemble $(\mathbf{dz}^n)^-$, $(\mathbf{dz}^n)^+$, $(\mathbf{dz}^{m-n})^-$, $(\mathbf{dz}^{m-n})^+$, J^n and $J^{(m-n)}$.
3. Factorize J^n and calculate $J^{(m-n)}(J^n)^{-1}$.
4. For each variable, $i=1,\ldots,n$, calculate $\mathbf{a}^i$, the ith row of $(J^n)^{-1}$ and carry out the maximization in (20) using a linear programming method. The resultant value of $\mathbf{a}^i.\mathbf{dz}^n$ is the ith element of $\mathbf{dx}^+$. Similarly, carry out the minimization in (21), to obtain the ith element of $\mathbf{dx}^-$.
5. Add $\mathbf{dx}^-$ and $\mathbf{dx}^+$ to $\hat{\mathbf{x}}$ to obtain $\mathbf{dx}^{1-}$ and $\mathbf{dx}^{1+}$.

Linear Programming method

Example

$$-1 < -x_1 + x_2 < 1$$
$$1 < x_1 + x_2 < 3$$
$$0.5 < \quad x_2 < 3$$



Linear Programming method

Example

$(\mathbf{z}^n)^l = [-1\ 1]^T$, $(\mathbf{z}^n)^u = [1\ 3]^T$, $(\mathbf{z}^{m+n})^l = [0.5]^T$, $(\mathbf{z}^{m+n})^u = [3]^T$,

$$J^n = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \text{ and } J^{m+n} = [0\ 1]$$

$$(J^n)^{-1} = \begin{bmatrix} -0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \text{ and } J^{m+n}(J^n)^{-1} = [0.5\ 0.5]$$

The cost function for i=1 $\quad \mathbf{a}^1.\mathbf{z}^n = -0.5z_1^{\,n} + 0.5z_2^{\,n}$

$(z_1^{\,n})^- = 1$ and $(z_2^{\,n})^- = 1$ $\qquad (z_1^{\,n})^+ = -1$ and $(z_2^{\,n})^+ = 3$

The cost function for i=2 $\quad \mathbf{a}^2.\mathbf{z}^n = 0.5z_1^{\,n} + 0.5z_2^{\,n}$

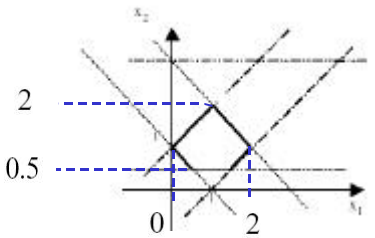
$(z_1^{\,n})^- = -1+s$ and $(z_2^{\,n})^- = 2-s$ $\qquad (z_1^{\,n})^+ = 1$ and $(z_2^{\,n})^+ = 3$

Linear Programming method

Example

Evaluating [x] we obtain

$$x_1^- = \mathbf{a}^1\mathbf{z}^{n-} = [-0.5\ 0.5][1\ 1]^T = 0,$$
$$x_1^+ = \mathbf{a}^1\mathbf{z}^{n+} = [-0.5\ 0.5][-1\ 3]^T = 2,$$
$$x_2^- = \mathbf{a}^2\mathbf{z}^{n-} = [0.5\ 0.5][-1+s\ 2-s]^T = 0.5$$
$$x_2^+ = \mathbf{a}^2\mathbf{z}^{n+} = [0.5\ 0.5][1\ 3]^T = 2,$$



Ellipsoid method

The linearised state uncertainty set is estimated by ellipsoids that have progressively reduced volume subject to problem constraints

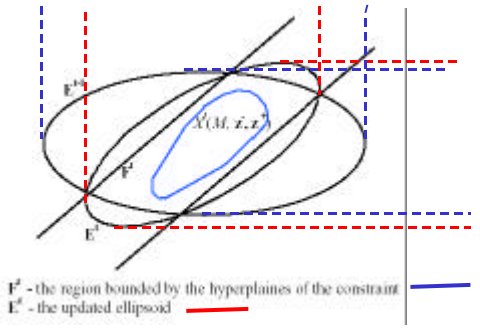
Ellipsoid

$$\mathbf{E}^t := \{\mathbf{x} \in \mathbf{R}^n : (\mathbf{x}-\mathbf{x}^t)^T P_t^{-1} (\mathbf{x}-\mathbf{x}^t) \leq 1.0\}$$

The linearised measurement constraints

$$\mathbf{z}^- - g(\hat{\mathbf{x}}) + J.\ \hat{\mathbf{x}} \leq J.\mathbf{dx} \leq \mathbf{z}^+ - g(\hat{\mathbf{x}}) + J.\ \hat{\mathbf{x}}$$

Ellipsoid method



Note that the shrinking ellipsoid can imply increase of bounds in some dimensions

Ellipsoid method

Using t^{th} constraint to update $t\text{-}1^{th}$ ellipsoid

$$\mathbf{x}^t = \mathbf{x}^{t-1} + (\rho_t v_t / (e_t^z)^2) P'_{t-1} . \mathbf{a}^t \quad (24)$$

$$P_t = (1 + \rho_t - (\rho_t v_t / ((e_t^z)^2 + \rho_t g_t))) P'_{t-1} \quad (25)$$

$$P'_{t-1} = (I + (\rho_t / ((e_t^z)^2 + \rho_t g_t)) P_{t-1} \mathbf{a}^t (\mathbf{a}^t)^T) P_{t-1} \quad (26)$$

$$g_t = (\mathbf{a}^t)^T P_{t-1} \mathbf{a}^t) \quad (27)$$

$$v_t = 0.5(z_t^+ + z_t^-) - (g(\mathbf{x}^{t-1}))_t + (J.\mathbf{dx})_t + (\mathbf{a}^t)^T.\mathbf{x}^{t-1} \quad (28)$$

The region F^t bounded by t^{th} constraint also contains the state uncertainty set $X^1([z])$

Ellipsoid method

Ellipsoidal confidence limit algorithm

1. Set t=0 and $P_0=\alpha I$.
2. Set $t=t+1$.
3. Calculate g_t and v_t from equations (27) and (28).
4. Find ρ_t that minimizes the volume of the new ellipsoid by solving the following quadratic equation in ρ_t:
   $(p-1)g_t^2 \cdot \rho_t^2 + ((2p-1)(e_t^z)^2 - g_t + v_t)g_t \cdot \rho_t + (e_t^z)^2(p(e_t^z)^2 - v_t^2) - g_t) = 0$
5. Calculate $P'_{t,I}$ from (26).
6. Update the state variable $\mathbf{x}^t$, as per equation (24).
7. Update P_t, equation (25).
8. If not all constraints have been processed yet than repeat from step 2.
9. If the volume of the ellipsoid has been reduced by less than a pre-specified ratio than stop, otherwise reset the constraints counter, t=0, and repeat from step 2.

Ellipsoid method

Example

$$-1 < -x_1 + x_2 < 1$$
$$1 < x_1 + x_2 < 3$$
$$0.5 < x_2 < 3$$

from the above we have

$\mathbf{z}^- = [-1\ 1\ 0.5]^T$ $\quad \mathbf{z}^+ = [1\ 3\ 3]^T$ $\quad \mathbf{x}^0 = [0, 0]$

$$J = \begin{bmatrix} -1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad P_0 = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

Ellipsoid method

Processing the constraints in order 1, 2, 3 we have

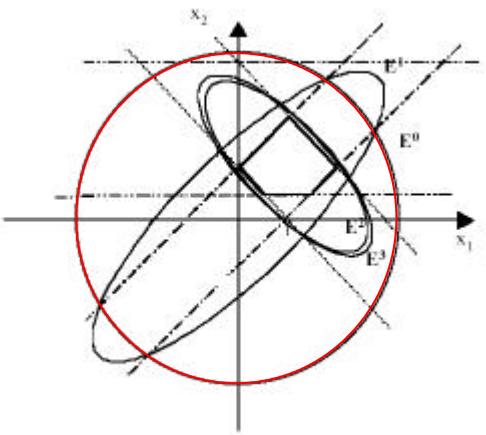
$g_1=20.00 \quad v_1=0.00 \qquad P_1=\begin{bmatrix} 7.26 & 5.09 \\ 5.09 & 7.26 \end{bmatrix} \qquad \mathbf{x}^{1-}=\begin{bmatrix} -2.69 \\ -2.69 \end{bmatrix} \qquad \mathbf{x}^{1+}=\begin{bmatrix} 2.69 \\ 2.69 \end{bmatrix}$

$g_2=24.70 \quad v_2=2.00 \qquad P_2=\begin{bmatrix} 2.18 & -0.21 \\ -0.21 & 2.18 \end{bmatrix} \qquad \mathbf{x}^{2-}=\begin{bmatrix} -0.62 \\ -0.62 \end{bmatrix} \qquad \mathbf{x}^{2+}=\begin{bmatrix} 2.33 \\ 2.33 \end{bmatrix}$

$g_3=2.18 \quad v_3=0.89 \qquad P_3=\begin{bmatrix} 2.48 & -0.18 \\ -0.18 & 2.48 \end{bmatrix} \qquad \mathbf{x}^{3-}=\begin{bmatrix} -0.74 \\ -0.30 \end{bmatrix} \qquad \mathbf{x}^{3+}=\begin{bmatrix} 2.41 \\ 2.44 \end{bmatrix}$

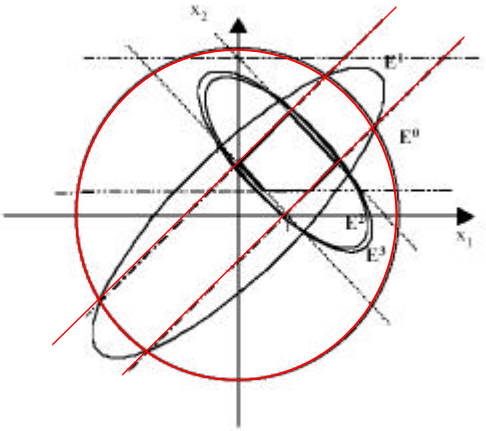
Ellipsoid method

Processing the constraints in order 1, 2, 3 we have



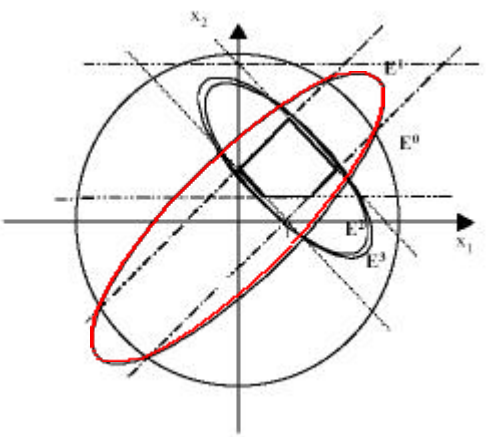
Ellipsoid method

Processing the constraints in order 1, 2, 3 we have



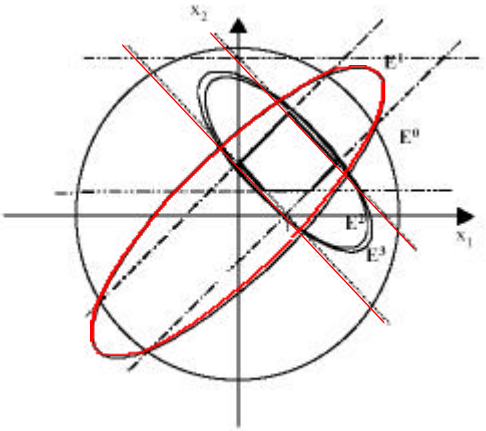
Ellipsoid method

Processing the constraints in order 1, 2, 3 we have



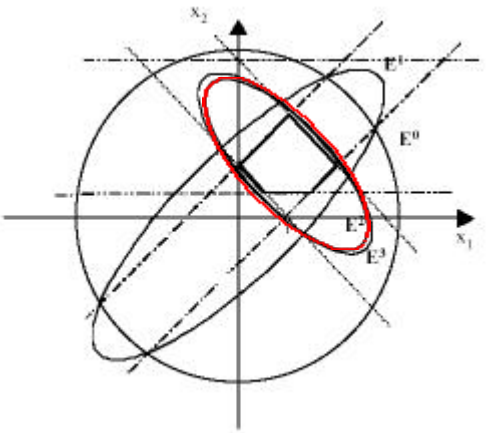
Ellipsoid method

Processing the constraints in order 1, 2, 3 we have



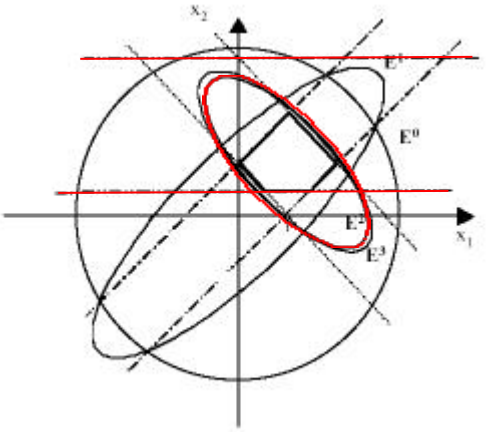
Ellipsoid method

Processing the constraints in order 1, 2, 3 we have



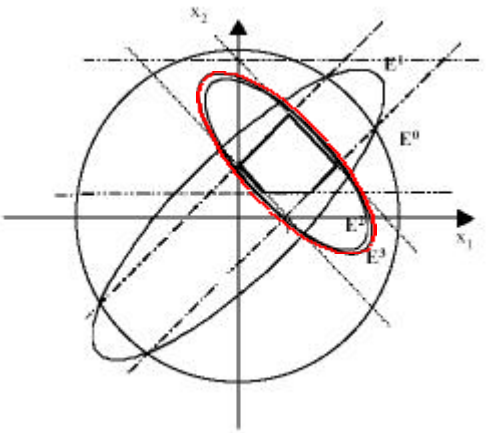
Ellipsoid method

Processing the constraints in order 1, 2, 3 we have



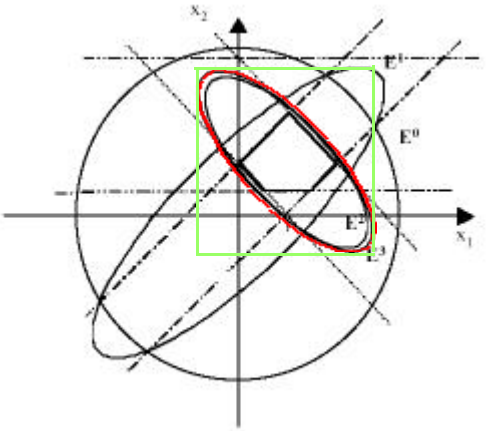
Ellipsoid method

Processing the constraints in order 1, 2, 3 we have



Ellipsoid method

Processing the constraints in order 1, 2, 3 we have



Ellipsoid method

Processing the constraints in order 3, 2, 1 we have

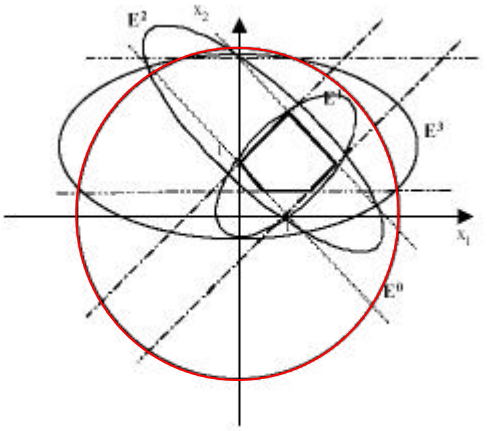
$g_3=10.00 \quad v_3=1.75 \qquad P_3=\begin{bmatrix} 12.33 & 0.00 \\ 0.00 & 3.12 \end{bmatrix} \qquad \mathbf{x}^{3-}=\begin{bmatrix} -3.51 \\ -0.46 \end{bmatrix} \qquad \mathbf{x}^{3+}=\begin{bmatrix} 3.51 \\ 3.07 \end{bmatrix}$

$g_2=15.44 \quad v_2=0.69 \qquad P_2=\begin{bmatrix} 5.46 & -2.51 \\ -2.51 & 3.25 \end{bmatrix} \qquad \mathbf{x}^{2-}=\begin{bmatrix} -1.89 \\ -0.38 \end{bmatrix} \qquad \mathbf{x}^{2+}=\begin{bmatrix} 2.78 \\ 3.22 \end{bmatrix}$

$g_1=13.73 \quad v_1=-0.97 \qquad P_1=\begin{bmatrix} 2.17 & 0.22 \\ 0.22 & 1.63 \end{bmatrix} \qquad \mathbf{x}^{1-}=\begin{bmatrix} -0.57 \\ -0.18 \end{bmatrix} \qquad \mathbf{x}^{1+}=\begin{bmatrix} 2.37 \\ 2.37 \end{bmatrix}$

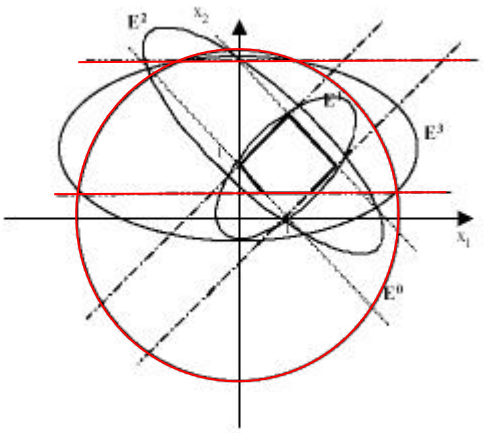
Ellipsoid method

Processing the constraints in order 3, 2, 1 we have



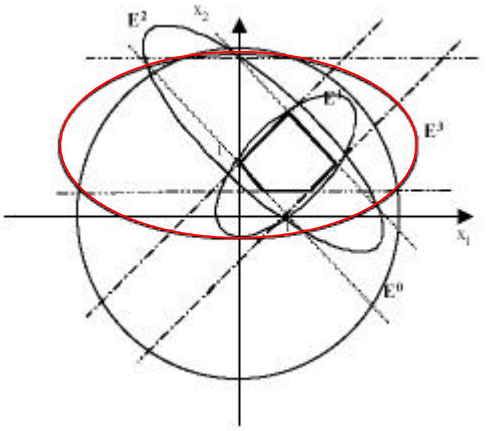
Ellipsoid method

Processing the constraints in order 3, 2, 1 we have



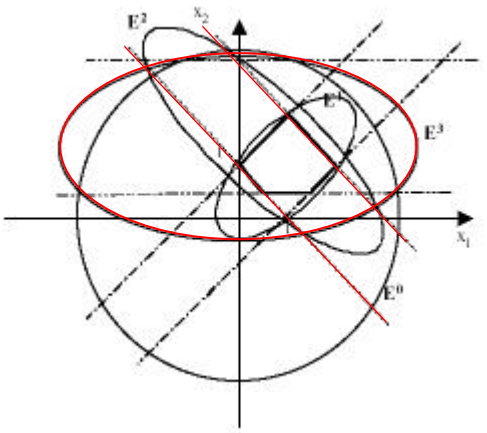
Ellipsoid method

Processing the constraints in order 3, 2, 1 we have



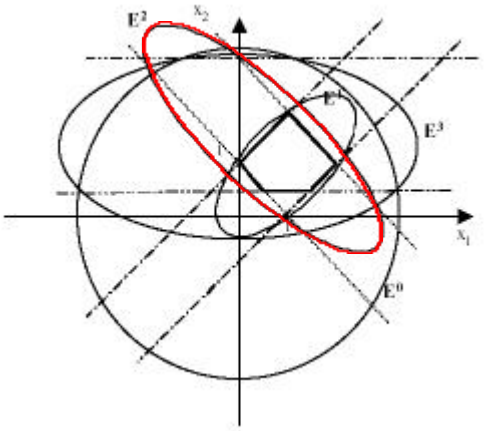
Ellipsoid method

Processing the constraints in order 3, 2, 1 we have



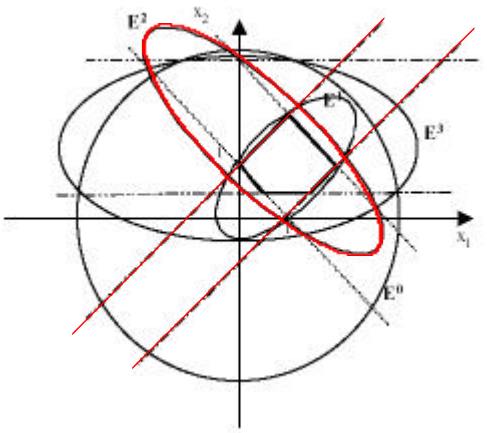
Ellipsoid method

Processing the constraints in order 3, 2, 1 we have



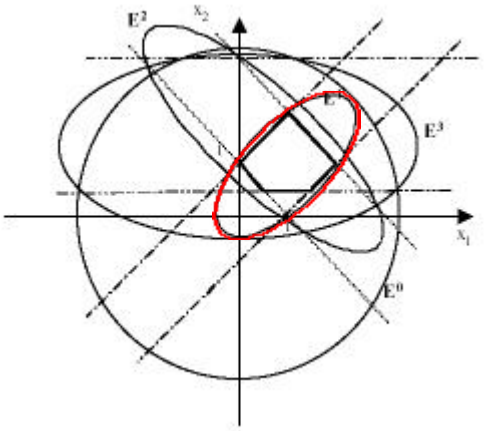
Ellipsoid method

Processing the constraints in order 3, 2, 1 we have



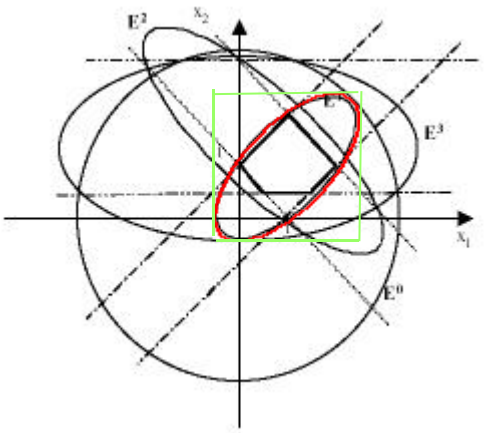
Ellipsoid method

Processing the constraints in order 3, 2, 1 we have



Ellipsoid method

Processing the constraints in order 3, 2, 1 we have



Sensitivity matrix method

The basis of the method is the observation that for a minimal measurement set the linearised uncertainty bounds can be calculated directly

However since we deal usually with the overdetermined measurement sets we calculate the pseudo-inverse of the Jacobean matrix – *sensitivity matrix* $\mathbf{dx} = (J^T J)^{-1} J^T.\mathbf{dz}$

The approximate linearised state uncertainty set is

$$X^2([\mathbf{z}]) := \{\mathbf{x} \in \mathbf{R}^n : \mathbf{x} = \hat{\mathbf{x}} + \mathbf{dx},\ \mathbf{dx} = (J^T J)^{-1} J^T.\mathbf{dz},\ \mathbf{dz} \in DZ([\mathbf{z}])\}$$

$$X^1([\mathbf{z}]) \subseteq X^2([\mathbf{z}])$$

Sensitivity matrix method

Sensitivity matrix confidence limit algorithm

1. Set $i = 0$.
2. Factorize the matrix $J^T J$ (This can be done using an augmented matrix formulation so as to preserve the condition number of the matrix J).
3. Set $i = i + 1$.
4. Calculate $\mathbf{b}^i$, the ith row of the sensitivity matrix $(J^T J)^{-1} J^T$ (This can be done efficiently taking into account the sparsity of J and using the augmented matrix based factorization of step 2).
5. Put $x_i^{2,+} = \mathbf{b}^i.\mathbf{dz}^+ + \hat{x}_i$, where $dz_j^+ = \begin{cases} dz_j^+ & \text{if } b_j^i > 0.0 \\ dz_j^- & \textit{otherwise} \end{cases}$
6. Put $x_i^{2,-} = \mathbf{b}^i.\mathbf{dz}^- + \hat{x}_i$, where $dz_j^- = \begin{cases} dz_j^- & \text{if } b_j^i > 0.0 \\ dz_j^+ & \textit{otherwise} \end{cases}$
7. If $i < n$, go back to step 3. Otherwise stop.

Sensitivity matrix method

Example

$$-1 < -x_1 + x_2 < 1$$
$$1 < x_1 + x_2 < 3$$
$$0.5 < x_2 < 3$$

$$J = \begin{bmatrix} -1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$$

The sensitivity matrix is

$$(J^T J)^{-1} J^T = \begin{bmatrix} -0.5 & 0.5 & 0 \\ 0.333 & 0.333 & 0.333 \end{bmatrix}$$

Sensitivity matrix method

Using interval arythmetic formalism we calculate bounds on every state variable using corresponding rows of the sensitivity matrix

$$[x_i] = [\mathbf{b}^i].[\mathbf{dz}] + [\hat{x}_i] \qquad , i=1,\ldots,n$$

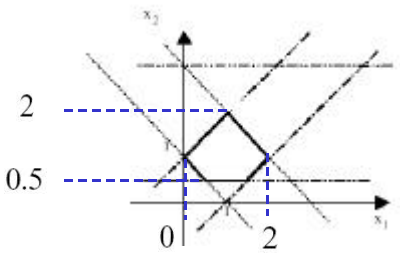
$$[\mathbf{b}^1]=[[-0.5, -0.5]\ \ [0.5, 0.5]\ \ [0, 0]]$$
$$[\mathbf{b}^2]=[[0.333, 0.333]\ \ [0.333, 0.333]\ [0.333, 0.333]]$$

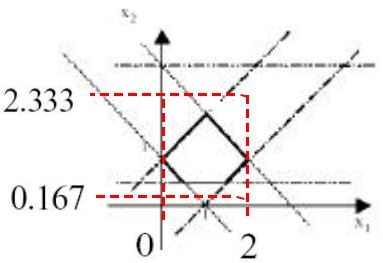
$$[\mathbf{z}]=[[-1, 1]\ \ [1, 3]\ \ [0.5, 3]]$$

Sensitivity matrix method

$[x_1] = [\mathbf{b}^1].[\mathbf{z}] = [-0.5, -0.5][-1, 1] + [0.5, 0.5][1, 3] + [0, 0][0.5, 3] =$
$= [-0.5, 0.5] + [0.5, 1.5] + [0, 0] = [0, 2]$

$[x_2] = [\mathbf{b}^2].[\mathbf{z}] = [0.333, 0.333][-1, 1] + [0.333, 0.333][1, 3] + [0.333, 0.333][0.5, 3] =$
$= [-0.333, 0.333] + [0.333, 1.0] + [0.167, 1.0] = [0.167, 2.333]$



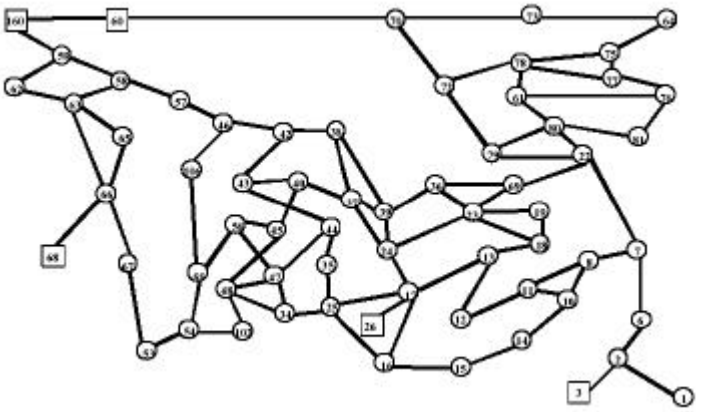


Sensitivity matrix method

The over-estimate of the state uncertainty set is due to the “inactive” constraints not being disregarded but taken into consideration in the sensitivity matrix calculation

Sensitivity matrix method

The real-life example

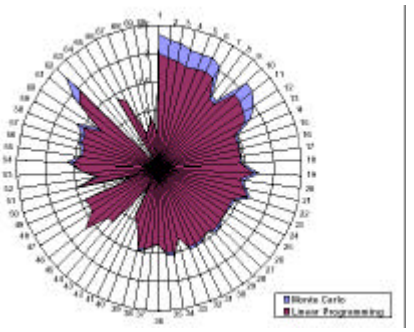


Sensitivity matrix method

State uncertainty variability sets (MC and LP)

$$\Delta X(M1, z^-, z^+) := \{ \delta x \in R^{+n} : \delta x = |dx|,\ \hat{x} + dx \in X(M1, z^-, z^+) \}$$

$$\Delta X^l(M1, z^-, z^+) := \{ \delta x \in R^{+n} : \delta x = |dx|,\ \hat{x} + dx \in X^l(M1, z^-, z^+) \}$$

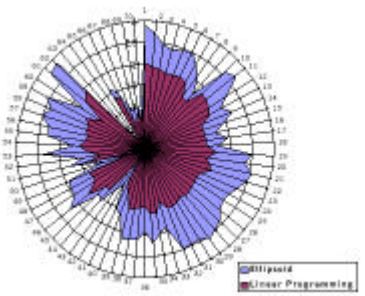


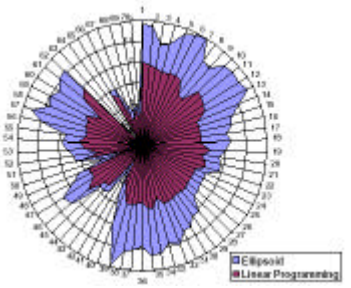
Sensitivity matrix method

State uncertainty variability sets (E and LP)

$$\Delta X^1(M2, z^-, z^+) := \{ \boldsymbol{\delta x} \in \mathbf{R}^{+n} : \boldsymbol{\delta x} = |\mathbf{dx}|,\ \hat{\mathbf{x}} + \mathbf{dx} \in X^1(M2, z^-, z^+) \}$$

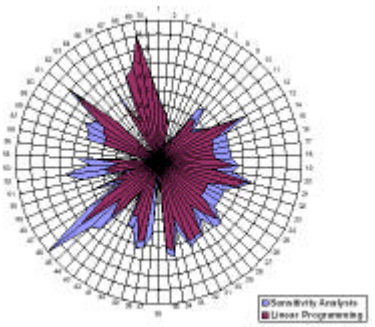
$$\Delta X^2(M2, z^-, z^+) := \{ \boldsymbol{\delta x} \in \mathbf{R}^{+n} : \boldsymbol{\delta x} = |\mathbf{dx}|,\ \hat{\mathbf{x}} + \mathbf{dx} \in X^2(M2, z^-, z^+) \}$$





Sensitivity matrix method

State uncertainty variability sets (SM and LP) evaluated with additional meters



Conclusions

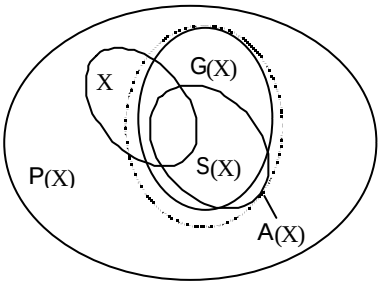
1. Monte Carlo estimation – accurate but computationally very expensive
2. Linear Programming estimation – accurate and more efficient than MC but still very demanding computationally
3. Ellipsoid estimation – very efficient computationally but produces conservative bounds that also depend on the order of processing of the constraints
4. Sensitivity Matrix estimation – accuracy comparable to LP and efficiency comparable to Ellipsoid method.

Information Structuring in Natural Language Communication: Syntactical Approach

Motivation

- A new framework for Natural Language processing
- NL processing and Data Mining
- NL processing and information granulation
- Stimulus for future research (validated on some simple NL examples)

Set representation of an NL



X – set of words;
P(X) – NL statements utilizing words X
G(X) – grammatically correct NL statements
A(X) – acceptably correct NL statements
S(X) – NL statements that are meaningful

Set representation of an NL

- Syntactic analysis
  - Statement is a member of A(X)
- Semantic analysis
  - Statement is a member of S(X) (conveys a specific information arising from the purposeful combination of words into sentences)

Syntactic processing roots

- Nativist view of cognition
  - Chomsky, *Cartesian Linguistics (1965)*
  - All NLs share certain universal structural features
  - Children acquire command of NL surprisingly fast regardless of the language
  - Justification of syntactical analysis of NL

NL processing

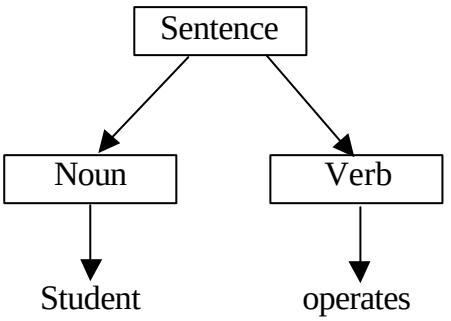
- Lexical acquisition
  - Naïve solution: lexicon construction (inflexible, error-prone, domain dependent)
  - Corpora: annotated language samples as a source of information and statistics (step towards resolving ambiguity and uncertainty)
  - From crisp (Homenda, 1991) to fuzzy relationships (defined on noun-phrases) as a means of capturing contextual information.

NL processing

- Syntactical analysis
  - Problems with NL ‘compilation’
  - Need for the flexibility in application of the rules of grammar
  - Baseline: sensible application of the context free grammars

NL processing

- Syntactical analysis



NL processing

- Context free grammars

  (CFG proposed here generates a language that includes only a subset of English)

- Tuplet G=(V, T, P, S) is CFG

  V – finite set of variables

  T – finite set of terminal symbols

  S – beginning symbol of the grammar

  P – finite set of productions (pairs)

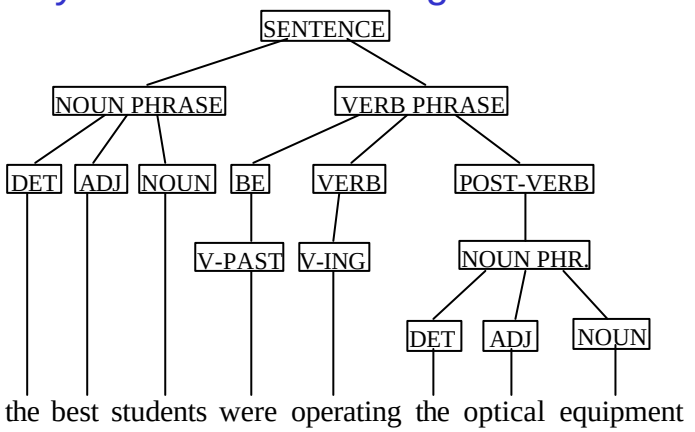
NL processing

- Example

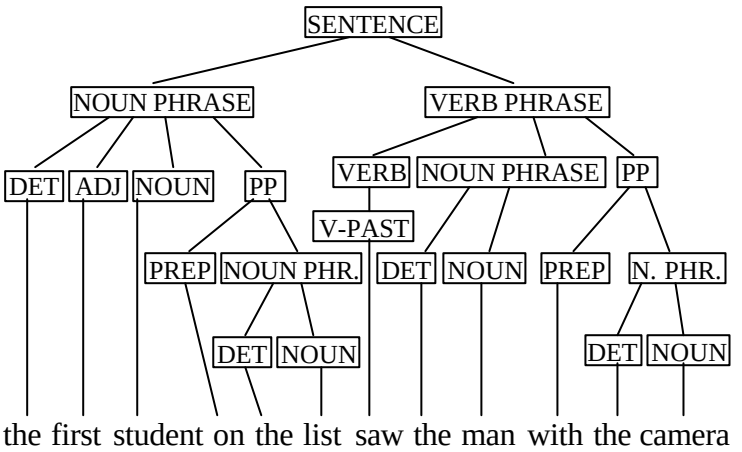
G = ({“Sentence”,“Noun”,“Verb”},
{student, operates},
{“Sentence” -> “Noun” “Verb”,
“Noun” -> student
“Verb” -> operates },
“Sentence”)

Parsing NL

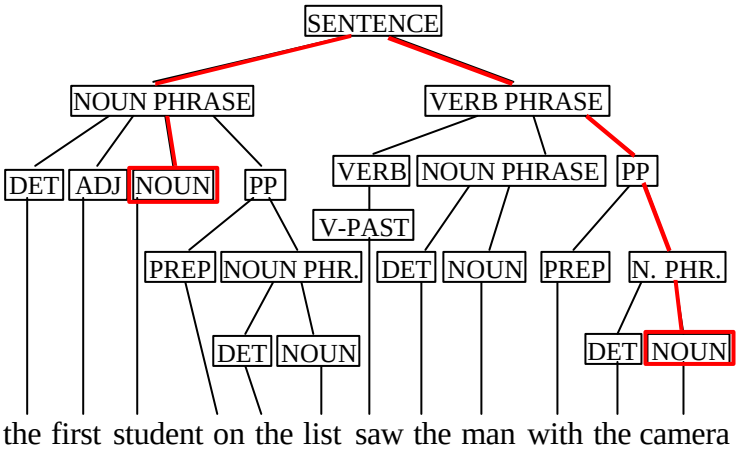
- Syntactical structuring



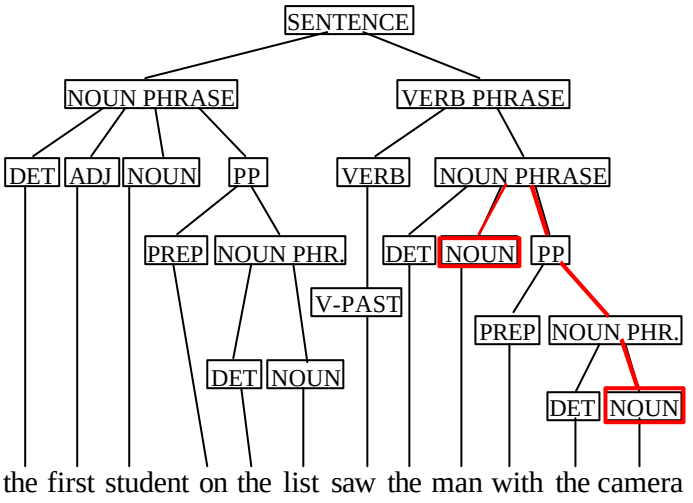
Parsing NL (ambiguity)



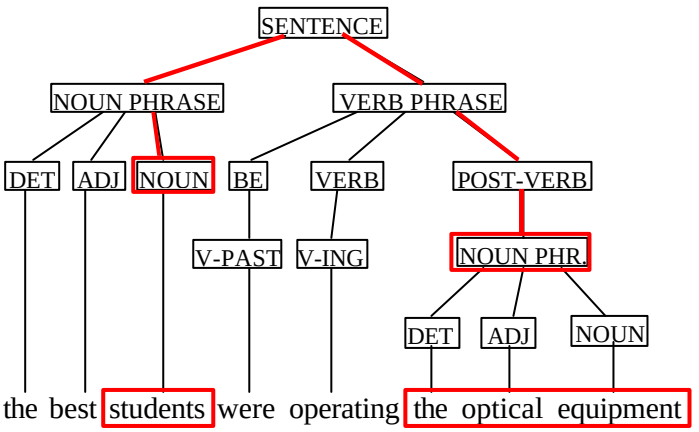
Parsing NL (ambiguity)



Parsing NL (ambiguity)

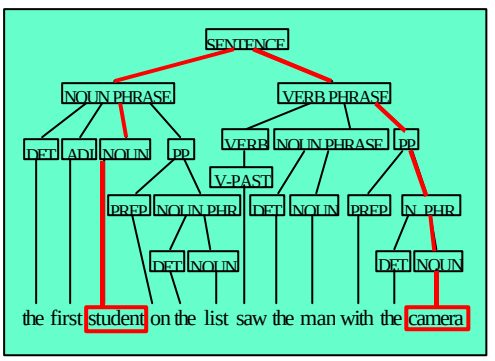


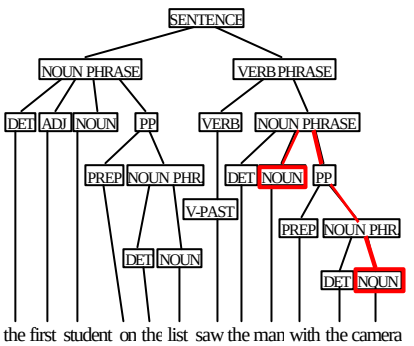
Information granulation
(resolving ambiguity)



Information granulation
(resolving ambiguity)

Lexicon information:
Camera – optical equipment





Dynamic lexicon

fs(camera) = { ... , 0.9/camera, 0.7/equipment, 0.7/optical, 0.5/student, 0.5/man, ...}

fs(tcamera) = { ... , 0.9/camera, 0.7/equipment, 0.7/optical, 0.9/student, 0.5/man, ... , 0.9/the best students, 0.9/the optical equipment, 0.5/the man, ... }

fs(tcamera) = { ... , 0.9/camera, 0.7/equipment, 0.7/optical, 0.9/student, 0.5/man, ... , 0.9/the camera, 0.9/the best students, 0.9/the optical equipment, 0.5/the man, ... 0.9/the first student on the list, ...}

Conclusions

- NL processing cast as FS/FL information processing
- Examples confirm feasibility
- Further research needed to investigate methodology for building lexicon of fuzzy sets.