Tensor Models and Electronic Noses

\[ x_{ijk} = \sum_{f=1}^{F} a_{if} b_{jf} c_{kf} + e_{ijk} \]

Thomas Skov

Quality and Technology - Department of Food Science, The Royal Veterinary and Agricultural University, Denmark

thsk@kvl.dk - www.models.kvl.dk

Outline

- Electronic Nose data
- Data analysis
- Tensor modelling
  - Data structure
  - Shifted data
- Results
- Conclusion
What is an Electronic Nose?
- Sensors
- Fingerprint

Electronic Nose Data
- Electronic sensor array → Pre-processing → Pattern recognition system → Licorice X

Electronic Nose data
- Sample × Time × Sensor
- Feature extraction

$S_{\text{max}}$, $S_{\text{feature}}$, $S_{\text{total}}$
Data analysis

Raw sensor data → Pre-processed data → Components → Reconstructed data

PRE-PROCESSING
- Baseline correction
- Transformation
- Feature extraction
- Centering/scaling

PROCESSING
- Tensor modelling (e.g. PCA, PARAFAC)
- Constraints
- Variable selection

POST-PROCESSING
- Visualisation
- Parameter estimation
- Rotation of parameters
- Concentrations

Raw sensor data
Pre-processed data
Components
Reconstructed data

Data structure

Traditional approach
- Feature extraction or unfolding the data
- Two-way data
- Bi-linear tensor models: PCA or PLS

New approach
- Keep internal data structure
- Three-way data
- Tri-linear tensor models: PARAFAC

Tensor modelling
Tensor modelling

**PCA - bilinear model,**

\[ x_{ij} = \sum_{f=1}^{c} t_{if} p_{jf} + e_{ij} \]

**PARAFAC - trilinear model,**

\[ x_{ijk} = \sum_{f=1}^{F} a_{if} b_{jf} c_{kf} + e_{ijk} \]

### Shifted data – in general

<table>
<thead>
<tr>
<th>pH</th>
<th>Pressure</th>
<th>Mass</th>
<th>Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1.0</td>
<td>110</td>
</tr>
<tr>
<td>2</td>
<td>2.4</td>
<td>1.1</td>
<td>121</td>
</tr>
<tr>
<td>3</td>
<td>7.5</td>
<td>0.9</td>
<td>112</td>
</tr>
<tr>
<td>4</td>
<td>8.0</td>
<td>1.1</td>
<td>123</td>
</tr>
<tr>
<td>5</td>
<td>6.9</td>
<td>145</td>
<td>0.9</td>
</tr>
<tr>
<td>6</td>
<td>6.5</td>
<td>1.0</td>
<td>154</td>
</tr>
<tr>
<td>7</td>
<td>6.5</td>
<td>1.0</td>
<td>154</td>
</tr>
<tr>
<td>8</td>
<td>6.6</td>
<td>1.0</td>
<td>152</td>
</tr>
<tr>
<td>9</td>
<td>8.9</td>
<td>0.9</td>
<td>127</td>
</tr>
<tr>
<td>10</td>
<td>7.0</td>
<td>0.9</td>
<td>120</td>
</tr>
</tbody>
</table>

Data obscured

### Examples
- Retention times in Gas Chromatography
- Sensor time profiles in Electronic Noses
Tensor modelling

- **No shifts**
  Differences are just in magnitude and a bi/tri-linear model handles this perfectly

- **Shifts**
  Bi/tri-linear model do not work

---

**Pre-processed sensor signal**

**Loadings**

- **Shifted sensor time profiles**
Tensor modelling

Data
Sample × Variable

PCA

\[ X = T P^T + E \]

Data
Sample × Time × Sensor

PARAFAC2

Only BTB the same (B can vary)

Model parameters

<table>
<thead>
<tr>
<th>Model</th>
<th>Mode A</th>
<th>Mode B (time)</th>
<th>Mode C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unfold PCA</td>
<td>Score plot (I)</td>
<td>Loading plot ((J × K))</td>
<td></td>
</tr>
<tr>
<td>PCA</td>
<td>Score plot (I)</td>
<td>Loading plot ((K))</td>
<td></td>
</tr>
<tr>
<td>PARAFAC1</td>
<td>Score plot (I)</td>
<td>Sensor signal vs. time ((J × R))</td>
<td>Loading plot ((K))</td>
</tr>
<tr>
<td>PARAFAC2</td>
<td>Score plot (I)</td>
<td>Sensor signal vs. time ((J × K × R))</td>
<td>Loading plot ((K))</td>
</tr>
</tbody>
</table>

Hypothetical example: Data matrix: \(10 \times 100 \times 5\). Model with two components (\(R = 2\))

# parameters: Unfold PCA: 1020, PARAFAC1: 230, PARAFAC2: 1030
Results

- Score plots
- Improved clustering and separation using three-way models

PARAFAC1

PARAFAC2

Loadings of time dimension

Ph.D. Student Thomas Skov
The Royal Veterinary and Agricultural University, Denmark
thsk@kvl.dk - www.models.kvl.dk
Modelled data - sample

No model
Preprocessed data
PARAFAC2
Model with 2 components
PARAFAC1
Model with 3 components

Residuals - sensor

Shift in time profiles
PARAFAC1
PARAFAC2
Data structure is a key factor when analysing electronic nose data.

The potential problem of shifted data can be solved in the modelling step.

Higher order tensor models gives an improved representation of data.