

Latent Variable Methods Course

Learning from data

Instructor: Kevin Dunn
kevin.dunn@connectmv.com
<http://connectmv.com>

© Kevin Dunn, ConnectMV, Inc. 2011

Revision: 268:adfd compiled on 15-12-2011

Copyright, sharing, and attribution notice

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, please visit

<http://creativecommons.org/licenses/by-sa/3.0/>



This license allows you:

- ▶ **to share** - to copy, distribute and transmit the work
- ▶ **to adapt** - but you must distribute the new result under the same or similar license to this one
- ▶ **commercialize** - you are allowed to create commercial applications based on this work
- ▶ **attribution** - you must attribute the work as follows:
 - ▶ "Portions of this work are the copyright of ConnectMV", *or*
 - ▶ "This work is the copyright of ConnectMV"

We appreciate:

- ▶ if you let us know about **any errors** in the slides
- ▶ **any suggestions to improve the notes**
- ▶ telling us if you use the slides, especially commercially, so we can inform you of major updates
- ▶ emailing us to ask about different licensing terms

All of the above can be done by writing us at

courses@connectmv.com

If reporting errors/updates, please quote the current revision number: 268:adfd

Projects I

- ▶ Preferably combine it with your research (2 for 1)
 - ▶ Chapter/section of your thesis
 - ▶ Alternative way of looking at an existing data set
- ▶ Theoretical investigation
 - ▶ Cross-validation (e.g. data randomization)
 - ▶ Missing data handling alternatives
 - ▶ Robust PCA and PLS
 - ▶ Adaptive PCA and PLS (handles drift, disturbances)
 - ▶ Orthogonal signal correction (OSC)
- ▶ Many data sets on the internet; freely available
 - ▶ Kaggle.com data analysis competitions (win some money!)
 - ▶ Prediction credit score
 - ▶ Predict if a car will be a “kick” (bad purchase)
 - ▶ Predict when supermarket shoppers will next visit and how much they will spend

Projects II

- ▶ Your own data is always the most interesting. Some ideas:
 - ▶ Image analysis data: identifying defects reliably
 - ▶ Soft sensor development (e.g. distillation column). Open- vs closed-loop
 - ▶ Multiblock data analysis (e.g. lab data from multiple steps/instruments)
 - ▶ Control system performance: data from closed-loop systems to determine if performance has degraded
 - ▶ QSAR: review literatures and compare alternative approaches
 - ▶ Financial data: some examples freely available online.
- ▶ 1 page outline of ideas: 4 November, or earlier (email is OK)
- ▶ Class presentations of 15 minutes: 9 and 16 December 2011
- ▶ Report
 - ▶ printed version and PDF version
 - ▶ Due 9 January 2012 (tentative)
 - ▶ No more than 25 pages, all included.

Presentation expectations

- ▶ Should clearly state objectives
- ▶ Describe why you have selected preprocessing
- ▶ Any special pre-treatment to the data?
- ▶ Why PCA and/or PLS is appropriate to achieving your objective
- ▶ What was learned that was new?
- ▶ How was objective achieved with the model

- ▶ 12 minutes of slides
- ▶ 8 minutes of questions

Presentation dates

9 December

- ▶ Cheng
- ▶ Mudassir
- ▶ Harry
- ▶ Matthew
- ▶ Sharleen
- ▶ Caroline
- ▶ Ran
- ▶ Jake

16 December

- ▶ Brandon
- ▶ Yasser
- ▶ Rummana
- ▶ Lily
- ▶ Yanan
- ▶ Pavan
- ▶ Abdul

Why study batch systems?

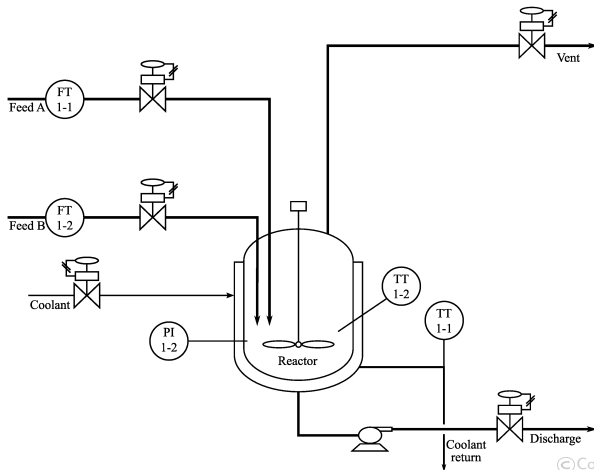
Batch manufacturing is widely used in many areas:

- ▶ bulk polymer manufacturing
- ▶ bulk food manufacturing
- ▶ fine chemicals
- ▶ “everyday” pharmaceutical drugs
- ▶ biological drugs
- ▶ semiconductor industry
- ▶ machining of tools and other manufactured parts

Usually found when operating on **small volumes** of *high-value product*.

A typical batch system

- ▶ Operators charge the reactor with material: *initial conditions*
- ▶ Additional materials (feed A and B) are added, sometimes manually, during the batch, but always according to a predetermined recipe

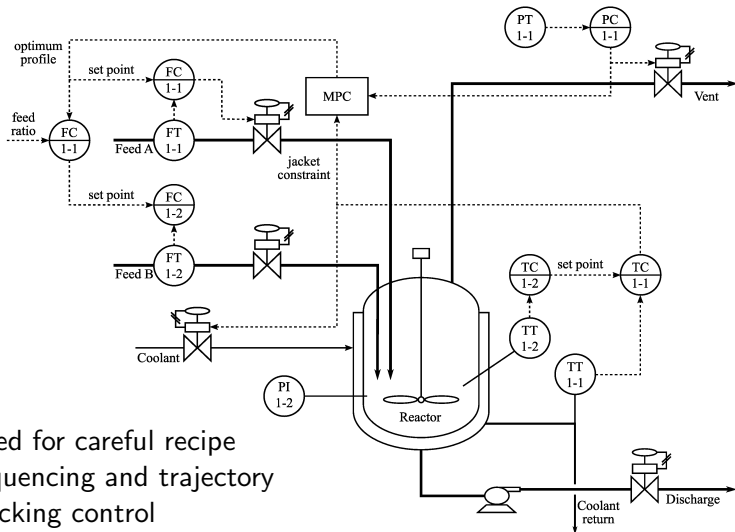


Batch systems: concepts and unique features

- ▶ Each batch is defined by its start time and end time
- ▶ The measured tags (variables) change between these points
 - ▶ In the above example: flow rates, temperature, and pressure

Batch systems with controllers

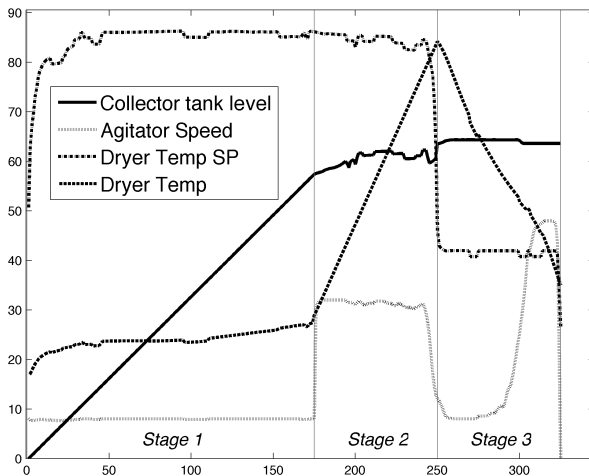
Batch systems can have very interesting control strategies:



Used for careful recipe sequencing and trajectory tracking control

Batch systems: concepts and unique features

- ▶ Usually have more than 1 phase (steps in the recipe)
- ▶ Nonlinear relationship between variables; and these relationships change with time, and between phases



Batch systems: terminology for these notes

N : number of batches

- ▶ literature sometimes uses I

K : number of tags

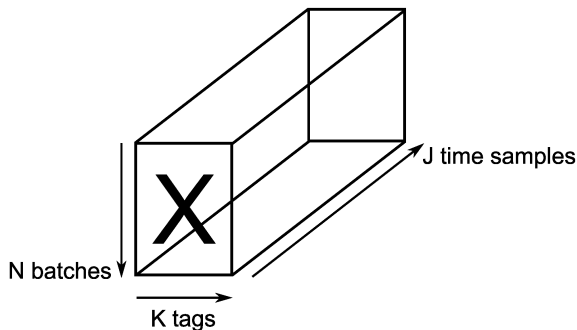
- ▶ literature sometimes uses J

J : number of time steps

- ▶ literature sometimes uses K

We aim for consistency with general latent variable methods:

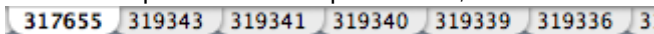
$$N \times K \times J$$



Batch systems: data representation

When retrieving batch data from computerized systems we often land up with:

1. One batch per sheet in a spreadsheet, with batch ID

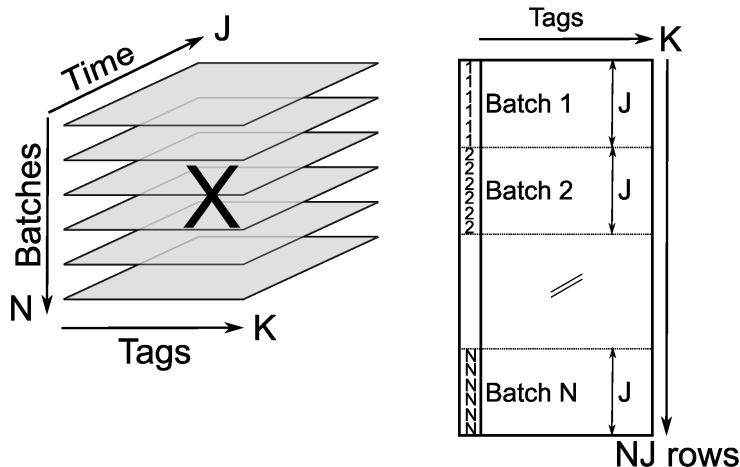


317655	319343	319341	319340	319339	319336	319333
--------	--------	--------	--------	--------	--------	--------

2. One batch per CSV file (we get N such files)

Batch systems: data representation

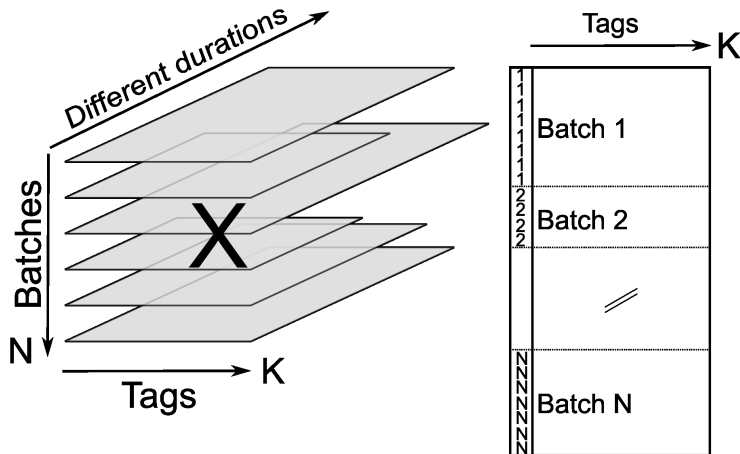
3. Stacked batches of **equal duration** in a single file



The first column is the batch ID: it is required for the software

Batch systems: data representation

4. Stacked, **but unaligned** batches, in a single file (common)



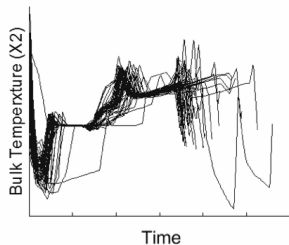
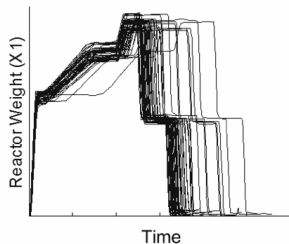
- ▶ Must include a batch ID column
 - ▶ Phased recipes: append a second “phase ID” column (not shown) within each batch
- © Conne

Batch systems: data representation

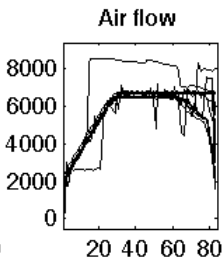
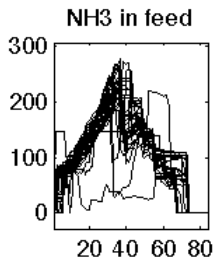
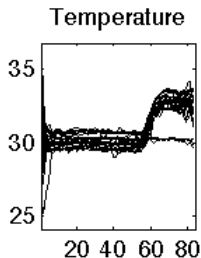
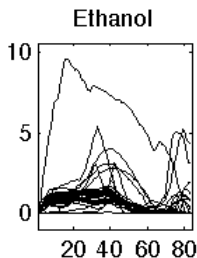
- ▶ It is very common that the time-dimension, J , is unequal
- ▶ We deal later with alignment: equalizing this time-dimension for all batches

Batch systems: examples of visualizing trajectory data

Unaligned data^a



Aligned data



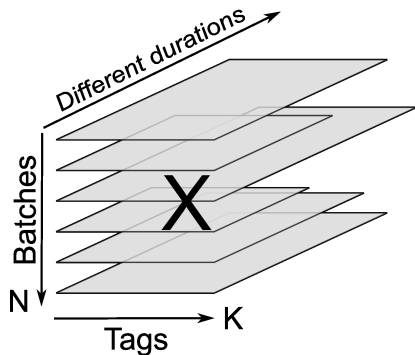
^aFrom [Cecilia Rodrigues' thesis](#) (used with permission)

There is substantial value in batch process data

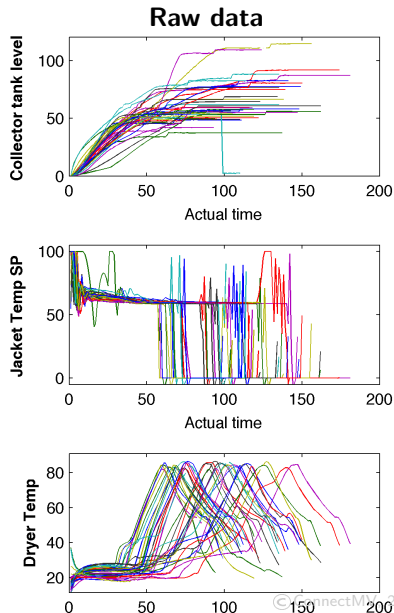
What can we achieve from our batch data?

1. Learning more / confirming relationships
2. Troubleshoot problems within a batch (*off-line*)
 - ▶ used to diagnose a recurring problem
 - ▶ then make permanent changes to fix it
3. Optimize and improve our trajectories for better CQAs
 - ▶ *Off-line*: used for recipe setpoints in future
 - ▶ *On-line*: called a “mid-course correction”
4. Realtime prediction of our
 - ▶ critical quality attributes (Y's)
 - ▶ batch endpoint
5. Realtime batch monitoring:
 - ▶ to detect abnormal batches, or
 - ▶ confirm a batch progressed as expected
 - ▶ Allows for quick release to next stage without waiting for lab results

Batch analysis using feature extraction



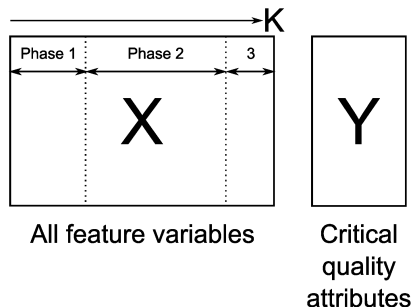
This is the case for most batch systems. Analysis of this raw batch data can be challenging.



Feature extraction methods: overview

- ▶ Use the same concept we saw earlier for steel surface images, knee acoustic data, flame images:
- ▶ *Extract features* from the raw batch data (next slide)

- ▶ Assemble the features: one row per batch
- ▶ Build an ordinary PCA on \mathbf{X}
- ▶ or if you have CQAs, then build a PLS: $\mathbf{X} \xrightarrow{\text{PLS}} \mathbf{Y}$



Major advantage: There is no need to align the batch data (we will look at alignment later)

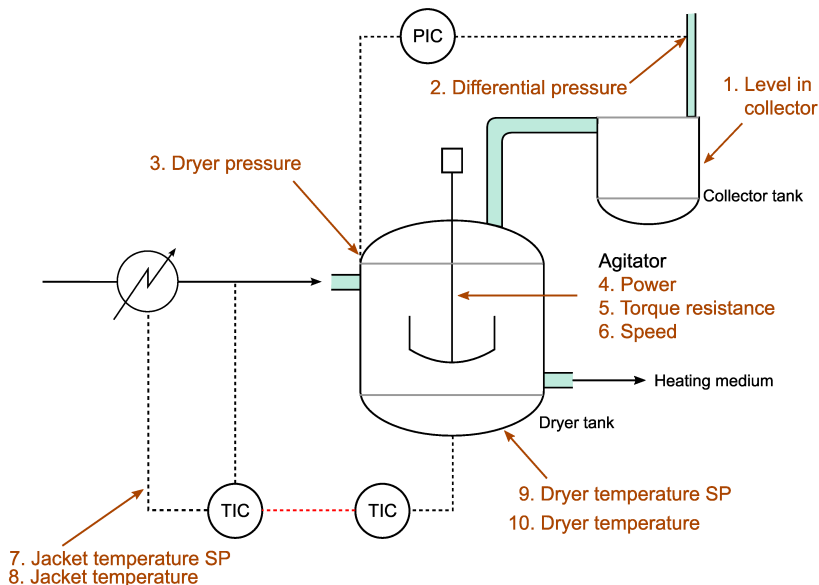
Which features to extract

- ▶ Extract features *within each phase* or within *specified windows*
- ▶ Extract for one, some, or **all** tags:
- ▶ average value ← easiest, often most useful feature
- ▶ median value
- ▶ integrated area under a tag (often makes engineering sense)
- ▶ standard deviation, (useful if tag should be constant)
- ▶ slope of curve
- ▶ energy or mass balance calculation over a phase
 - ▶ e.g. heat released by a reaction should be taken up by the cooling water
 - ▶ an imbalance can point out a problematic batch
- ▶ value of a trajectory at start/end of phase
- ▶ total time for a phase to complete

How to use feature-based models

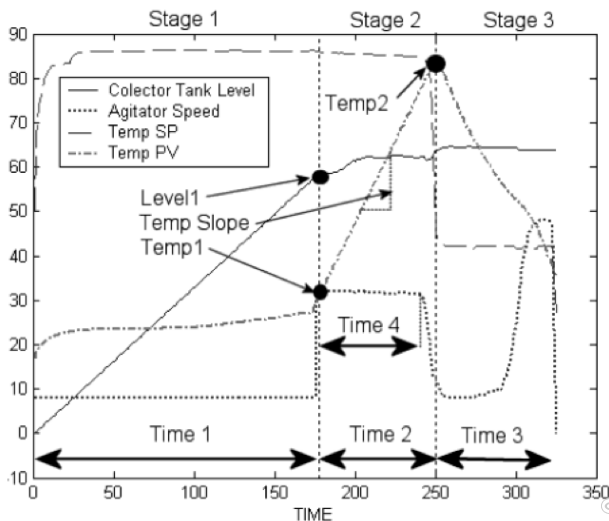
- ▶ Many features extracted: often ~ 80 to 100 columns; many are not useful
- ▶ Used in an ordinary PCA or PLS. All the usual tools apply:
 - ▶ VIP: finds most important variables for explaining CQAs (\mathbf{Y})
 - ▶ loadings and weights: used to interpret relationships between variables
 - ▶ use contributions between points and clusters in the scores
 - ▶ use R^2 per variable and small weights to decide which on less-useful features to eliminate
- ▶ Learn about the batch
e.g.: high standard deviation in cooling water temperature correlated with poor CQA value
- ▶ Troubleshooting an unusual batch
- ▶ By extension, use troubleshooting information to improve future batches
- ▶ Can be used for monitoring: we'll come back to this

In-class example



In-class example: using feature-based models

- ▶ Feature example for this batch process works well
 - ▶ 3 distinct phases with “sharp” trajectories
 - ▶ Features already extracted in the data file



In-class example: using feature-based models

- Many features extracted, but only these were useful:

<i>Name-phase</i>	<i>Calculated feature</i>	<i>Phase</i>
Level-Mean-3	Mean collector tank level	3
DiffPress-Mean-1	Average differential pressure	1
DryPress-Mean-1	Average drier pressure	1
Power-Slope-1	Slope of the power trajectory	1
Torque-Slope-1	Slope of the torque trajectory	1
DTemp-SP-Mean-3	Average drier temperature set point	3
DTemp-Mean-1	Average drier temperature	1
DTemp-Slope-2	Drier temperature slope	2
Temp1	Drier temperature at the end of phase 1	1
Temp2	Drier temperature at the end of phase 2	2
Time1	Time duration of phase 1	1
Time2	Time duration of phase 2	2
Time3	Time duration of phase 3	3

- **X**: the 13 extracted features
- **Y**: 8 final quality variables

In-class example: model results

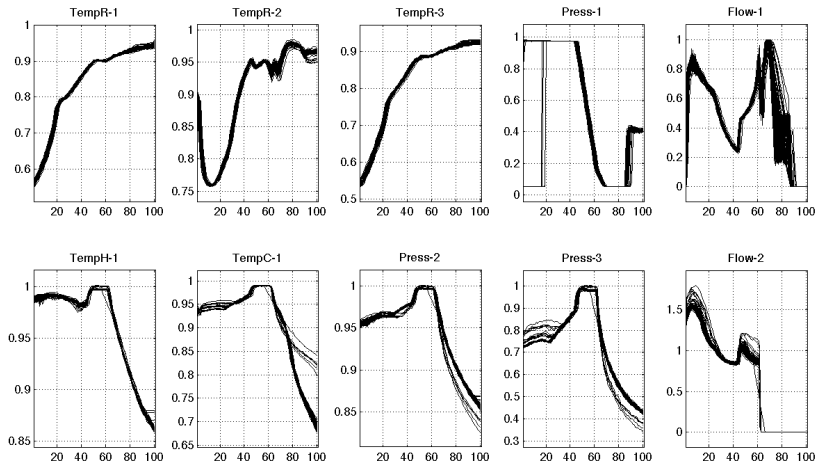
- ▶ Build a PLS model using the 13 features in the \mathbf{X} matrix
- ▶ Exclude the columns labelled \mathbf{Z}_i (we will come back to these later)
- ▶ Let $Y = [Y_1, Y_2, Y_4, Y_6, Y_9, Y_{10}, Y_{11}, \text{SolventConc}]$

Questions:

1. Show how “on-spec” and “off-spec” batches separate in the score plots
2. Use a contribution plot to understand why off-spec batches occur
3. Use the w^*c biplot to see the relationship between the variables and the “on-spec” / “off-spec” batches
 - ▶ Confirm, using the raw data, the relationships you see in the weights biplot
4. How would you tell operators to run future batches to ensure more “on-spec” batches?

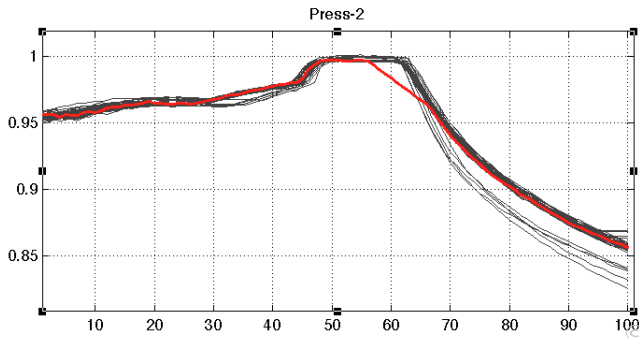
Disadvantages of feature-based models

- ▶ Smoothly varying trajectories within a phase
 - ▶ no distinct features and hard to quantify



Disadvantages of feature-based models

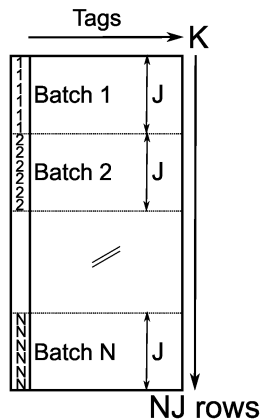
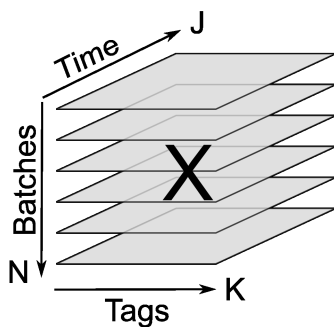
- ▶ Features to choose depend on engineer's prior knowledge
- ▶ Also on how you plan to use the feature-based model:
 - ▶ for off-line learning, or process optimization
 - ▶ for monitoring?
- ▶ Subtle defects and broken correlations between variables are often hard/impossible to detect and quantify
- ▶ The feature may in fact not exist in an abnormal batch: so it's missed



Dealing with batch data

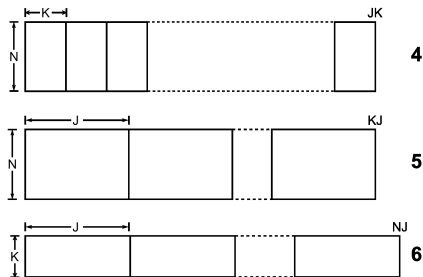
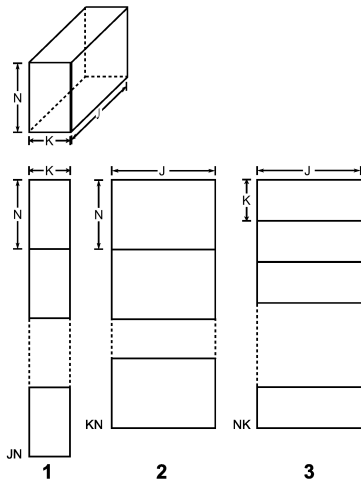
We will assume here that the batch data are pre-aligned.

- ▶ We will cover alignment later
- ▶ But we need to first see *what we want* from the models, so we can understand *how* to align



Unfolding a multiway data cube

- ▶ There are 3 ways to slice a cube (vertical, horizontal, depth)
- ▶ Then stack the slices either *side-by-side* or *top-to-bottom*
- ▶ There are 6 unique **combinations** of slicing **and** stacking
- ▶ For PCA: combinations **2** and **3** are the same, as are **4** and **5**



How to choose an appropriate unfolding

We will come back to this topic after seeing some examples.

Unfolding direction depends on

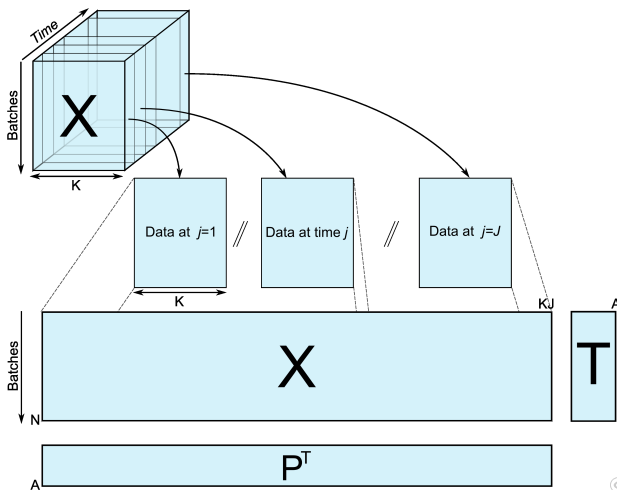
- ▶ what preprocessing will remove from the raw data
- ▶ what we want from model:
 - ▶ learn about relationships between rows – require one batch per row
 - ▶ learn about (cor)relation between columns – tags over time

In this respect, unfolding by method **4** or **5** is most appropriate.

Analysis and learning from batch data

Approach

Unfold the aligned batch data row wise, one batch per row. Then use PCA in the usual way.



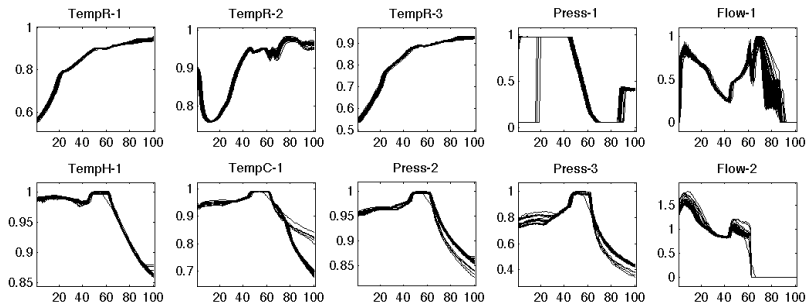
DuPont Nylon example: learning from new data

- ▶ Industrial data set of $N = 55$ batches from Nylon production¹
- ▶ Temperature, pressure and flow rate variables: $K = 10$ batch tags
- ▶ Batch duration about 2 hours ($J = 100$ time intervals)
- ▶ 12 hours elapse before lab values available
 - ▶ so batch-to-batch adjustment not possible
 - ▶ long hold-up times before determining disposition
- ▶ Known problems with batches: 38, 40, 41, 42, 50, 51, 53, 54, 55

¹More details can be found in [Paul Nomikos' PhD thesis](#)

DuPont Nylon example: raw data

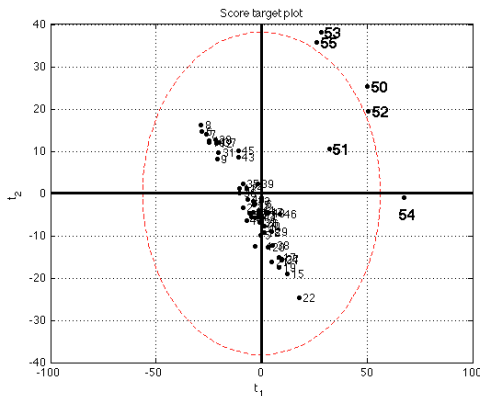
Note: data are scaled for confidentiality



- ▶ Can see a few unusual batches: see “Temp-C1” and “Press-1” tags
- ▶ Alignment looks pretty good (process is well controlled)
- ▶ Some periods are noisy: “Flow-1” and “Flow-2”

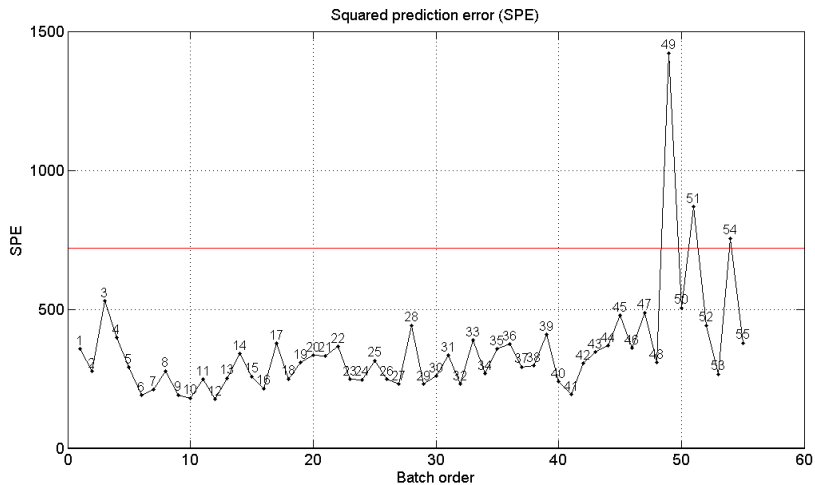
DuPont Nylon example: initial PCA

- ▶ Just start with 2 components initially
 - ▶ no cross-validation, just get a “feel” for the data
 - ▶ $R_X^2 = [38.3\%, 17.6\%]$, or cumulatively: 55.9%



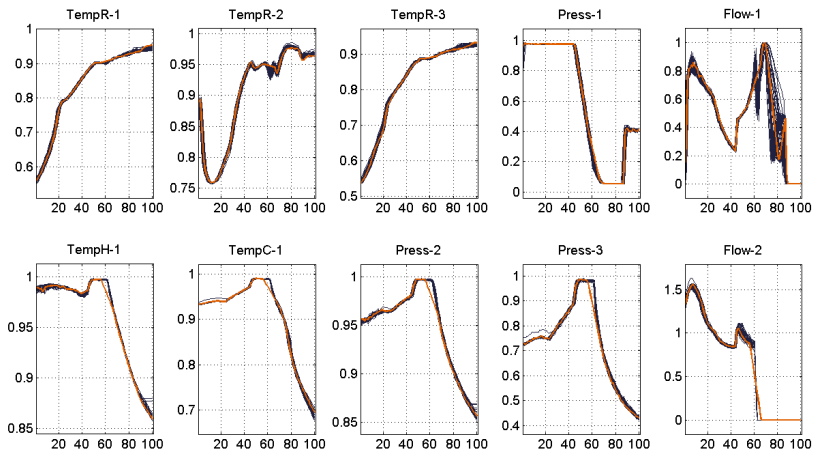
- ▶ Batches 50 to 55 unusual
- ▶ They distort the model
- ▶ Before excluding them and rebuilding model, let's first examine them (skip past a few slides).

DuPont Nylon example: SPE



DuPont Nylon example: Raw data for batch 49

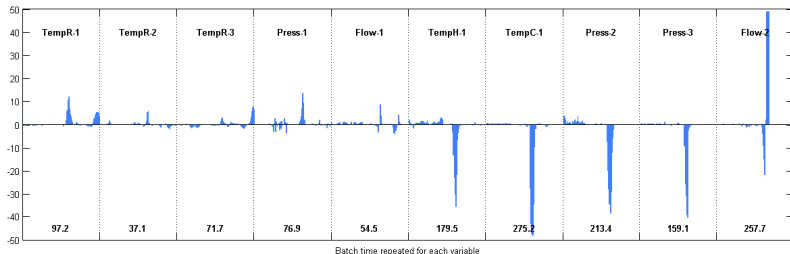
- If we try to examine the raw data for batch 49 directly, we may (wrongly) conclude FLOW-1 as cause of the problem



- But the true cause is given by the SPE contributions (next)

DuPont Nylon example: SPE

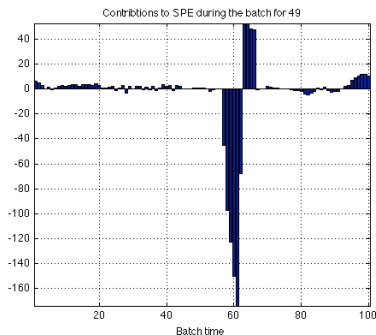
- ▶ Batch 49 shows in the overall SPE plot
- ▶ Let's look at the contribution plot
 - ▶ Contribution value for every tag (K), at every time step (J)
 - ▶ i.e. there are $JK = 1000$ contributions values
- ▶ Easy to visualize when grouped by tags (see below)
- ▶ Shows *when* the problem occurred: $t = 55$ to 67
- ▶ Shows *which* are the main tags



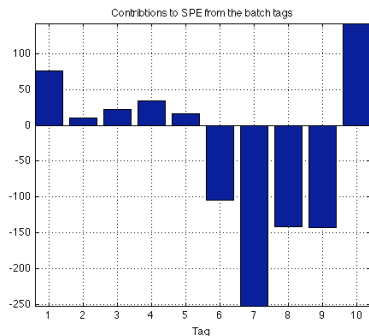
Confirm contributions in the raw data plots (red line in previous slide)

DuPont Nylon example: SPE summary

Sum contributions for all tags at every time step $J = 1 \dots 100$



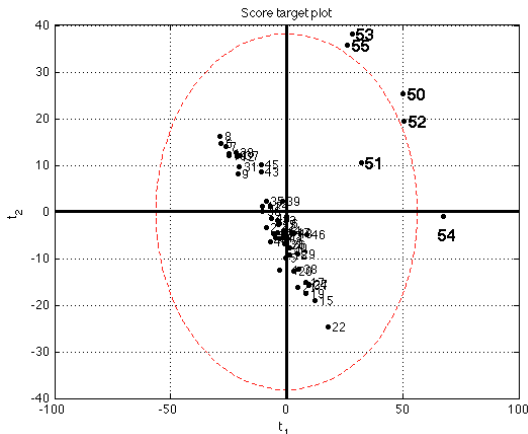
Sum contributions across all time steps for every tag $K = 1 \dots 10$



Cause seems to be due to small deviations in heating and cooling system (broken correlation) and pressure system. Nomikos reports this batch had barely acceptable final quality.

DuPont Nylon example: examining score outliers

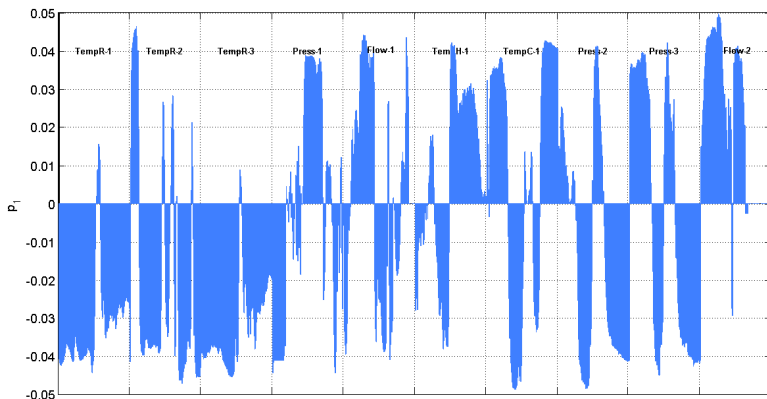
Batches 50 to 55 are all unusual (outliers) given the context of the remaining batches



We will examine the score contributions individually (*interactively in class*). We interpret these contributions in the same way as the SPE contribution (previous slide).

DuPont Nylon example: batch 54 (high t_1 batch)

We will examine the \mathbf{p}_1 loading for observation 54 instead (we will get a similar conclusion from contribution plot for batch 54)



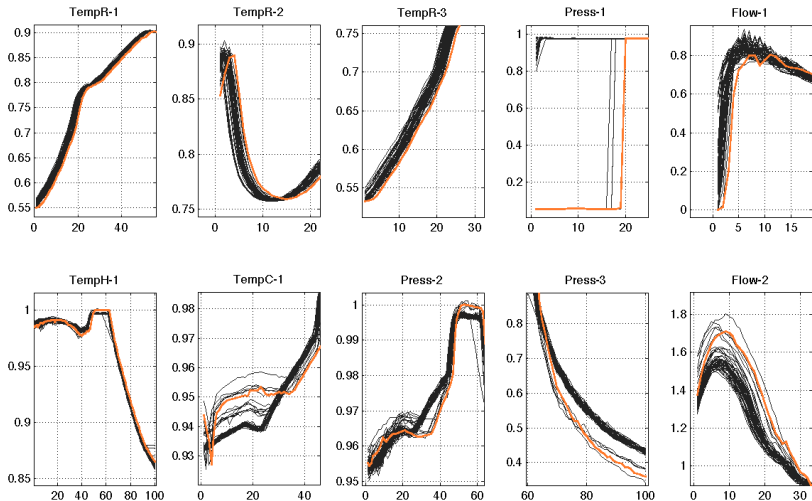
- As before, there are $JK = 1000$ columns in the bar plot

DuPont Nylon example: batch 54

- ▶ Batch 54 has a large positive t_1 value
- ▶ How to get a high t_1 value?
 - ▶ $(+x)(+p)$
 - ▶ $(-x)(-p)$
- ▶ So what was abnormal in the trajectories for batch 54?
 - ▶
 - ▶
 - ▶

DuPont Nylon example: batch 54 is marked

Confirm our findings in the raw data trajectories

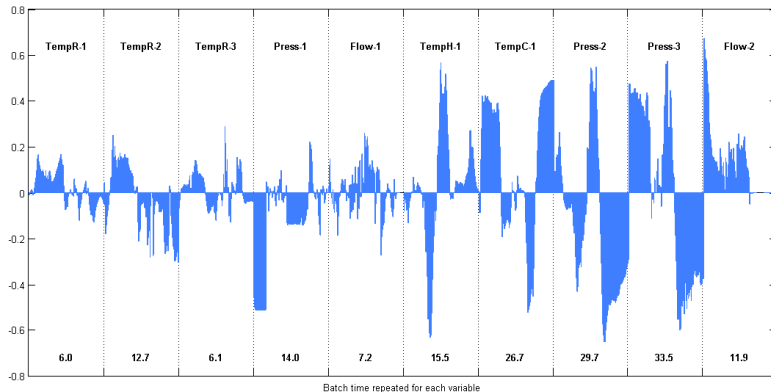


As can be seen, this same problem occurs with several batches.

Noticeable from raw data alone – in hindsight.

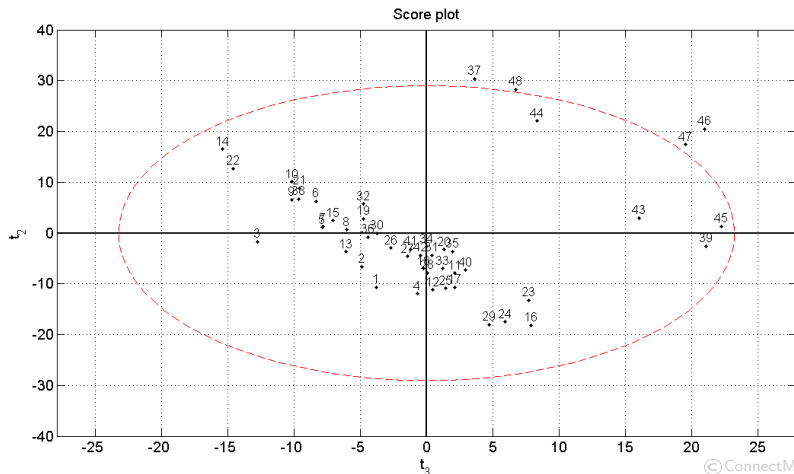
DuPont Nylon example: batch 55

Contribution plot for batch 55



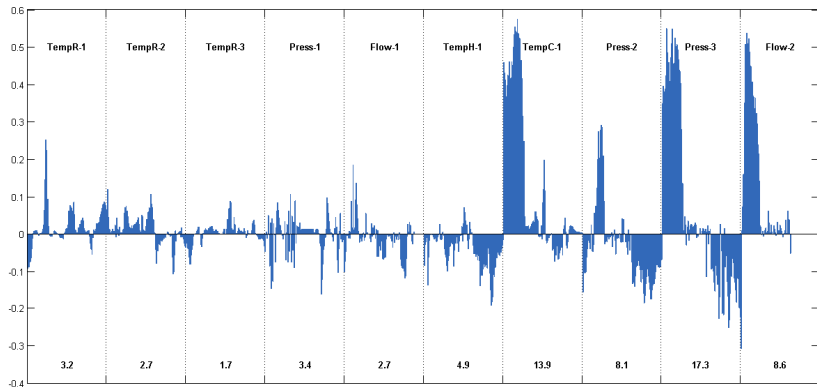
DuPont Nylon example: Exclude and rebuild

- ▶ Exclude batches 49, 50, 51, 52, 53, 54, and 55.
- ▶ Rebuild PCA model.
- ▶ We see another cluster show up in $t_2 - t_3$
- ▶ We will investigate those interactively in class



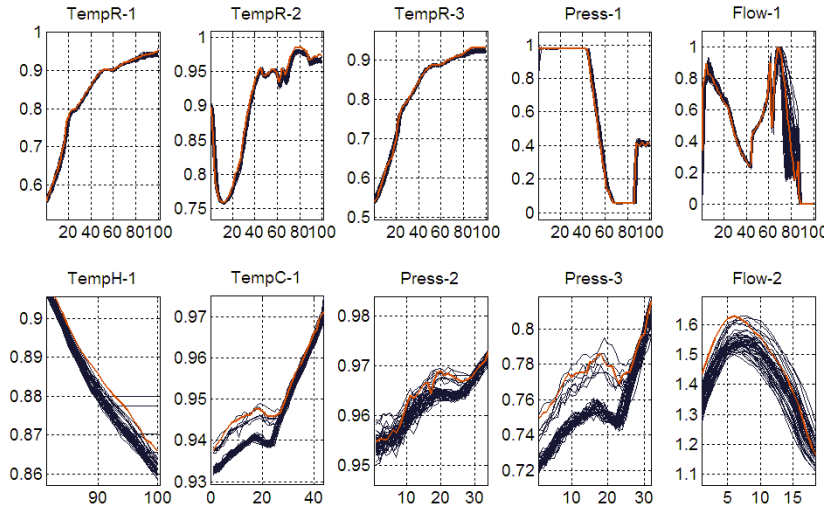
DuPont Nylon example: batch 39

Contribution plot for batch 39



Numbers in bold are the integrated sum for each tag.

DuPont Nylon example: batch 39 raw data

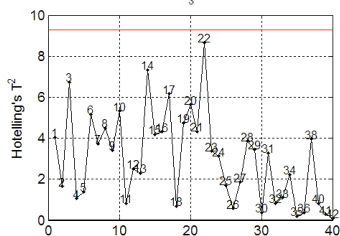
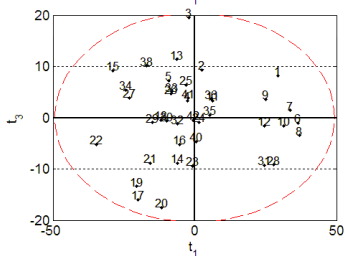
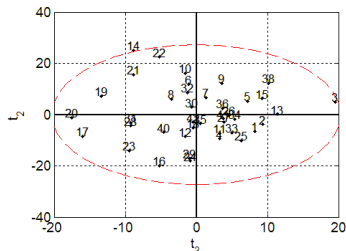
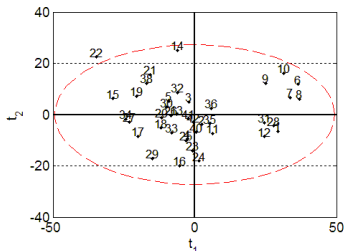


Batch 39 (and 37, 43, 44, 45, 46, 47, 48) were **not bad** batches. Just operated in a different way: but not to cause poor quality.

DuPont Nylon example: final model

Additional batches were excluded (37, 39, 43, 44, 45, 46, 47, 48) and a third model built. More even distribution of scores.

Investigate why there is a group of batches with high t_1 scores.



DuPont Nylon example: Note

- ▶ Batches 38, 40, 41 and 42 were known to have problematic critical quality (CQAs)
- ▶ No cause/detection can be found in the plots for these batches
- ▶ Problems are not present in trajectories
 - ▶ perhaps we aren't measuring an important trajectory
 - ▶ perhaps due to some aspect of the raw materials added?
- ▶ **Key point:** the measurements must contain the information required to make a classification (*observability requirement*)

What have we learned so far

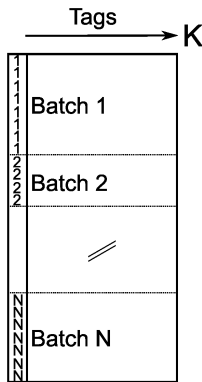
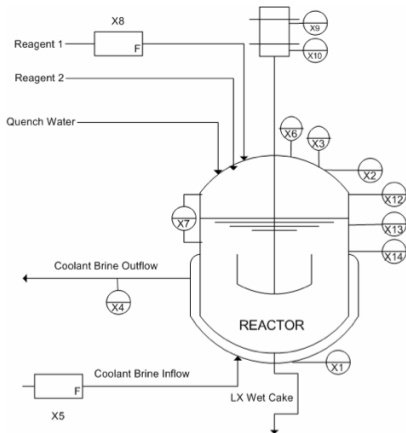
1. How to interpret a batch loadings plot
 - ▶ Loading plots show “*loading evolution over time*” for each tag
 - ▶ We also see how tags are correlated
 - ▶ **Very important point:** those relationships will change *during* the batch
2. Contributions to scores, SPE, or T^2
 - ▶ Can be shown grouped by tag
 - ▶ Summed over all tags for a given time point: “*time-evolution*”
 - ▶ Summed over all time for a given tag: “*tag-by-tag*” contribution
3. Similarly for VIPs

What have we learned so far

4. We can calculate R^2 for each column
 - ▶ Group R^2 values, and show “ R^2 evolution over time” for tag
 - ▶ Also can calculate R^2 over *all time points* for a given tag
 - ▶ Calculate R^2 over *all tags* for a given time point: shows “ R^2 evolution over time” for the model
 - ▶ Finally, calculate R^2 for entire unfolded \mathbf{X} : this is the usual R^2 value

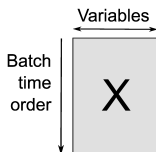
Batch systems: simple definition

A collection of the **same variables (tags)**, gathered over a **period of time**, for a **number of batches**.



Batch systems: changing relationships over time

Batch system: group of the *same variables*, gathered over a period of *time*. Why not just use ordinary PCA or PLS?



- ▶ PCA is invariant to row order:
 - ▶ can shuffle rows, still get same model
 - ▶ PCA explains how rows are related to each other
 - ▶ each row is assumed independent of the others
 - ▶ summarizes relationship between variables (columns)

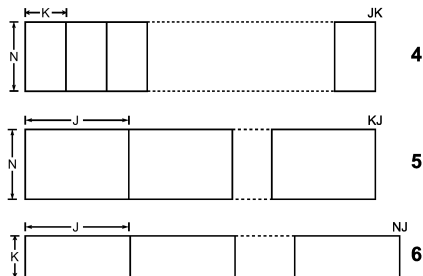
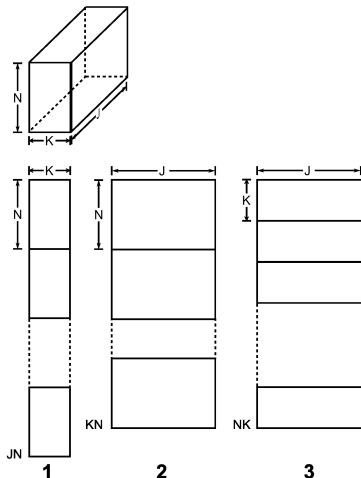
Key issue

In **batch systems**: relationship between variables **changes during the batch**. So the above approach is not appropriate.

Return back to unfolding

We showed earlier that:

- ▶ There are 6 unique **combinations** of slicing **and** stacking
- ▶ For PCA: combinations **2** and **3** are the same, as are **4** and **5**



How to choose an appropriate unfolding

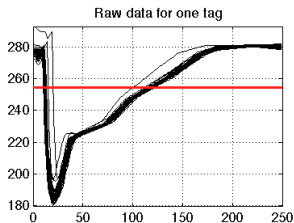
We said earlier that unfolding method chosen should depend on:

1. What preprocessing will remove from the raw data
 - ▶ we will compare unfolding method 1 and (4/5)
 - ▶ Method **1**: called observation-wise unfolding (OWU)
 - ▶ Method **4** or **5**: called batch-wise unfolding (BWU)
 - ▶ Next slide shows the effect of centering on 1 tag from a batch system
 - ▶ Variance of the tag, before centering: 715.6 units^2
 - ▶ Then we show variance of residuals after fitting a PCA model on OWU data

Unfolding: two main approaches compared

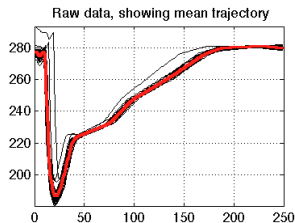
Unfolding method 1

Observation-wise unfolding

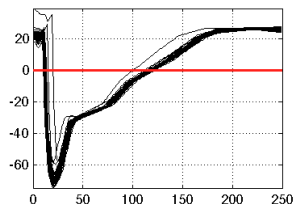


Unfolding method 4

Batch-wise unfolding

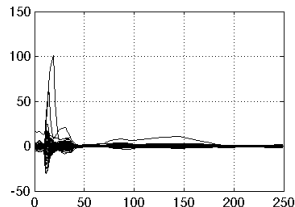


Unfolding method 1: JN by K matrix, then centering



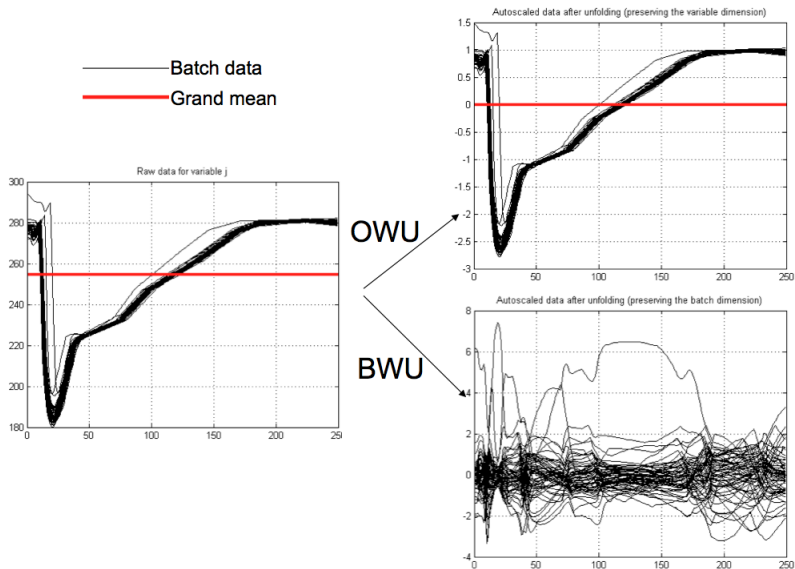
Variance to explain: 715.6 units²

Unfolding method 4: N by JK matrix, then centering



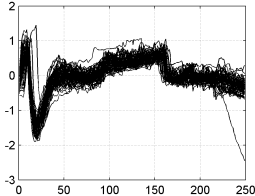
Variance to explain: 9.571 units²

Unfolding, then centering and scaling

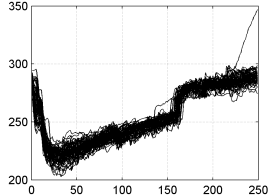


Unfolding: using OWU (unfolding method 1)

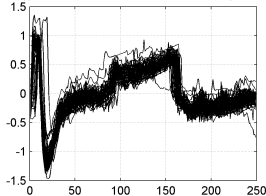
(d) OWU: residuals after the first component



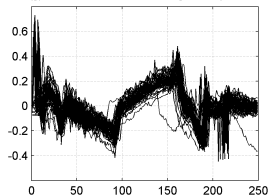
(e) Predictions (from OWU) with one component



(f) OWU: residuals after the second component



(g) OWU: residuals after fitting 8 components



(f) Variance to explain: 71.5 units²

(g) Variance to explain: 13.1 units²

OWU required 8 components to reach a comparable variance to BWU with zero components.

How to choose an appropriate unfolding

Unfolding method also depends on:

2. What we want from the latent variable model:

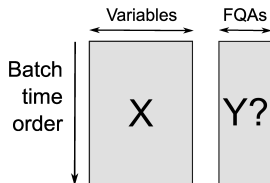
- ▶ to learn about relationships between batches – we require one batch per row
 - ▶ to learn about correlation between tags over time – put these in the columns
-
- ▶ In this respect, unfolding by method **4** or **5** (BWU) is most appropriate.
 - ▶ BWU captures the non-linearities which are always present in batch systems
 - ▶ The loadings capture the changing correlation between the K tags over the J time steps

Batch systems: changing relationships over time

1. Relationship between variables change within a batch
 - ▶ Entry and exit temperatures in a batch dryer correlate (move together) at the end of the batch.
 - ▶ But, at start of the batch there is little relationship: heat goes to evaporating moisture
 - ▶ Many other examples: usually correlation structures change from one phase to another
2. Past history of a batch affects future behaviour. By definition a batch is an “**integrating system**”
 - ▶ Unfolding batch data using method **1** (long, thin matrix) will not capture that “past effect” on future rows. Why?
 - ▶ Recall: each row is independent in PCA and PLS
3. Recall **lagging** from an earlier class:
 - ▶ We lagged data from earlier rows into the current row
 - ▶ This improved the model’s performance
 - ▶ Unfolding horizontally (method **4** and **5**) is just a form of “lagging” the entire batch history into 1 row

Batch systems: changing relationships over time

4. Predictive modelling is harder by observation-wise unfolding this way



- ▶ What would we use as the \mathbf{Y} matrix in this case? We don't have \mathbf{Y} values at each time step.
 - ▶ Many elaborate schemes proposed in literature to try force this structure. Why? Only advantage I can see is that we don't require alignment.
5. Simple solution: arrange data so that each batch is one row
- ▶ There are some issues with this (alignment, real-time monitoring)
 - ▶ In general, these issues can be addressed effectively

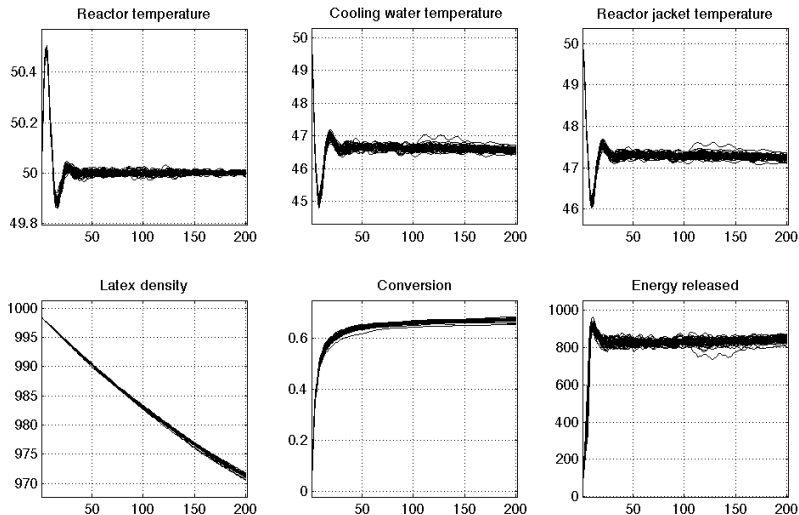
Batch PLS example: SBR

- ▶ Simulated data from *first principles* mechanistic model for styrene butadiene rubber²
- ▶ Simulations are useful to make sure models identify what we expect
- ▶ Simulation contained mostly “normal operating conditions”
 - ▶ 2 problematic batches were simulated
 - ▶ the same fault, but starting at different times
- ▶ **Y**-space quality variables:
 1. Composition
 2. Particle size
 3. Branching
 4. Cross linking
 5. Polydispersity

² More details can be found in [Paul Nomikos' PhD thesis](#)

SBR: raw data

- Batches data: $N = 53$; Tags: $K = 6$; Time steps: $J = 200$



SBR: build model

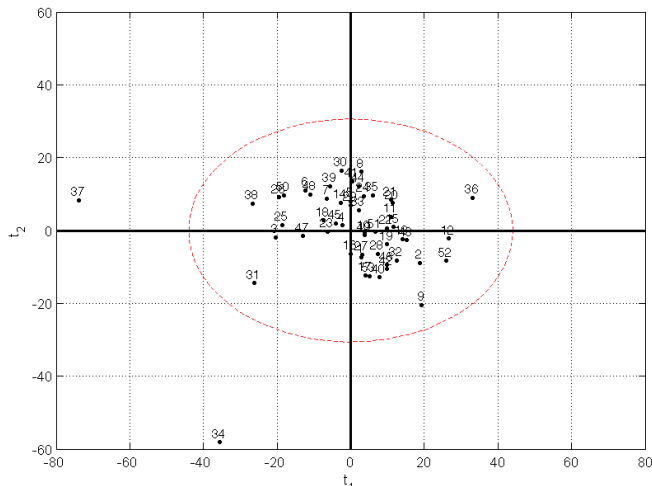
Approach:

- ▶ Normally I would start with a PCA on the **X**-space trajectories to understand the trajectory relationships
- ▶ Then a PCA on the **Y**-space quality variables to see if there are unusual batches
- ▶ In this data set: both these PCA models give the same interpretation as PLS
- ▶ So we only show the PLS results here.

PLS results:

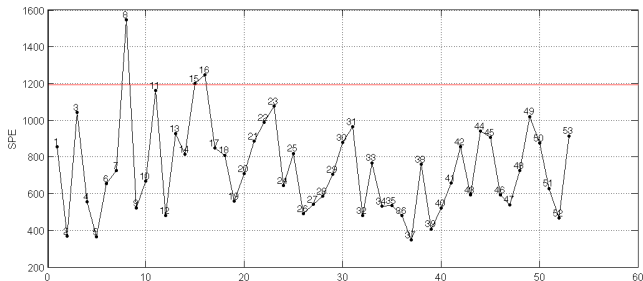
- ▶ Start with 2 to 3 components: *just to see what's going on*
- ▶ $R^2_{X,1} = 24.5\%$ and $R^2_{X,2} = 12.7\%$
- ▶ $R^2_{Y,1} = 65.3\%$ and $R^2_{X,2} = 6.9\%$
- ▶ Next: scores, weights, SPE, T^2 ... all the usual PLS tools

SBR: PLS score plot



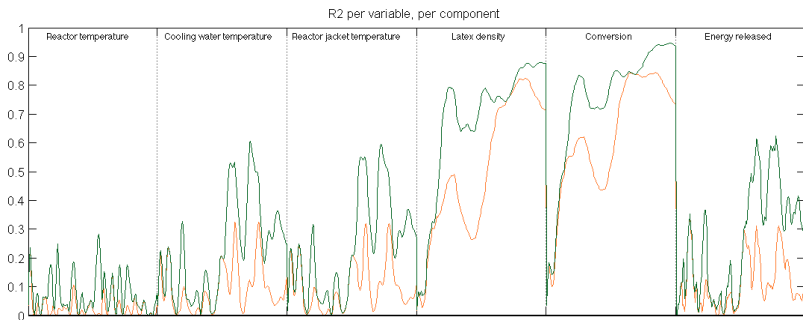
- Batches 34 and 37 were in fact the unsuccessful batches! This shows promise.

SBR: check SPE



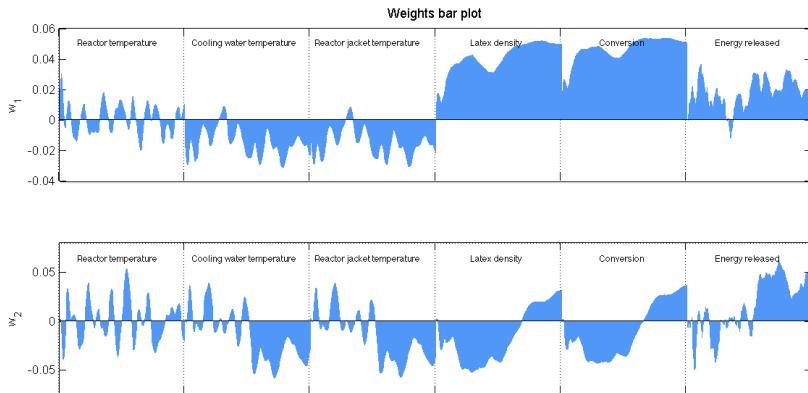
- No problems picked up. This is the overall SPE, using data from the *entire* batch.

SBR: understand R^2 breakdown in the \mathbf{X} -space



- ▶ LV1 and 2: latex density and conversion dominate the model
- ▶ R^2 is low at start because all batches are similar initially
 - ▶ after centering and scaling there is just noise at the start.

SBR: PLS weights

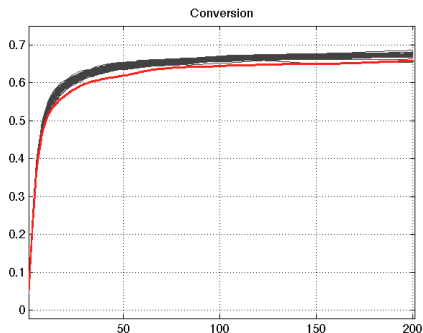
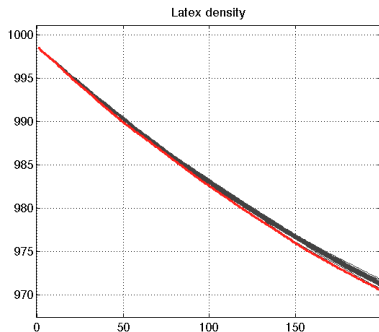


From the above we can infer that:

- ▶ batch 37 had low t_1 because of
 - ▶ **below average latex density** *throughout the batch*
 - ▶ **below average conversion** *throughout the batch*

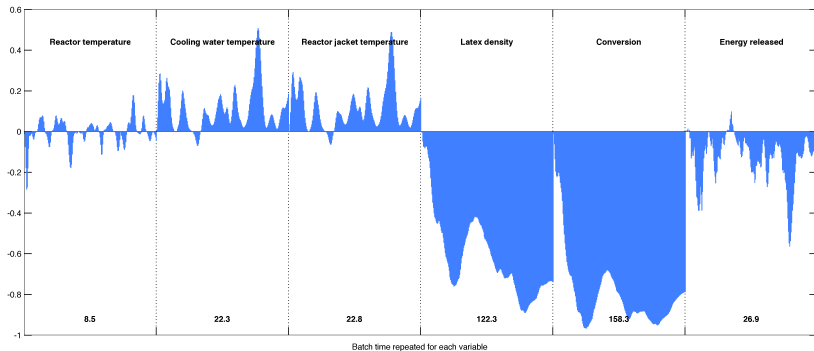
Confirmed in the raw data, and contribution plot for batch 37 ... next

SBR: raw data for batch 37 (to confirm)



- Confirmed our interpretation with the raw data
- **True cause** (from simulation): 30% greater organic impurity in butadiene feed, from the start of the batch

SBR: contribution plot for batch 37

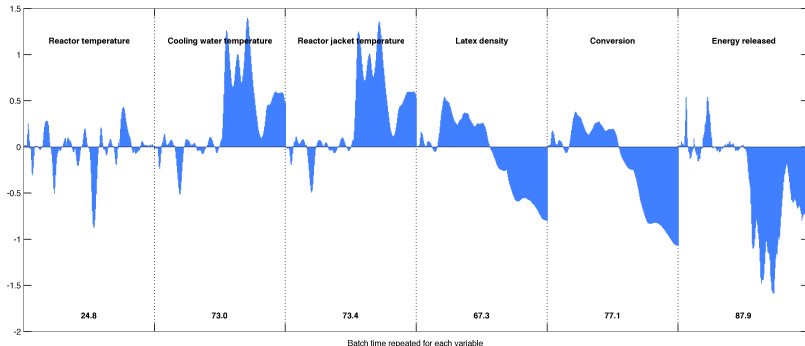


- Contributions highest for the latex density and conversion, as expected.

SBR: investigate batch 34

Batch 34 had high t_2 :

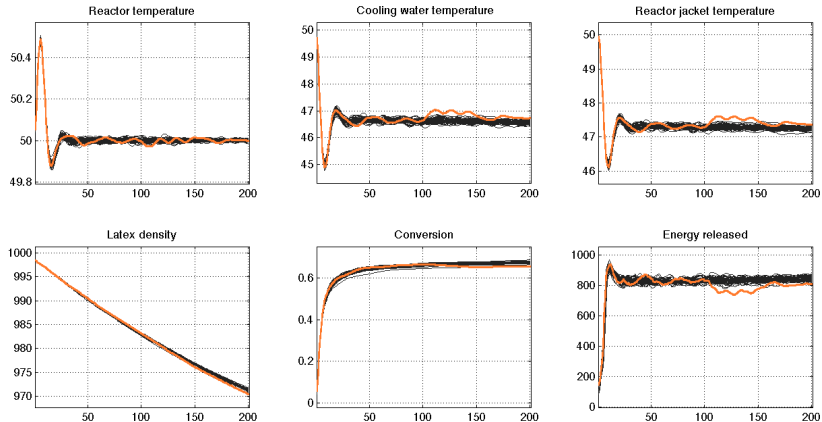
- ▶ From weights plot for w_2 (earlier): we expect the problem to be due to cooling water, jacket temperature, and below average energy released in last half of the batch
- ▶ Contribution plot confirms this:



This affected the density and conversion as well.

SBR: investigate batch 34

Raw data for this batch is highlighted



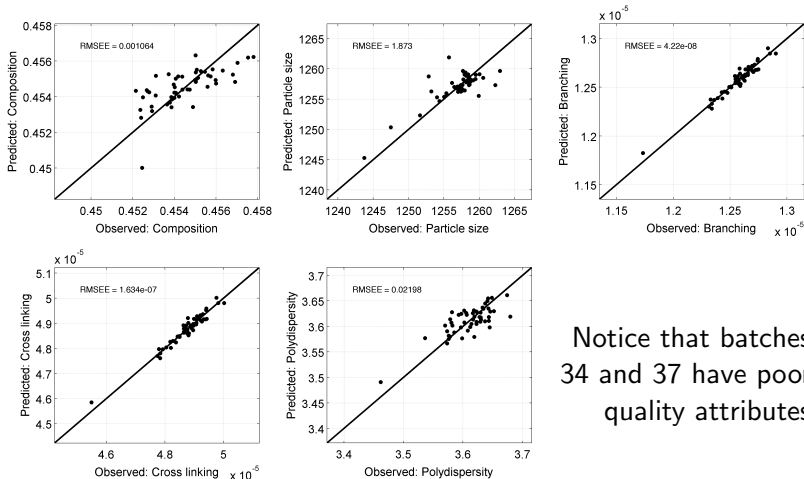
Confirms the problem occurred. Same problem as before, 30% greater organic impurity in butadiene feed, but only midway during the batch progress.

Interesting observation

- ▶ The same fault occurred in batch 34 and 37.
- ▶ But they show up in different locations in the score plot
- ▶ Because the *time when the fault occurred* is different

SBR: predictions from the model

We also get predictions from the batch PLS model for the 5 quality variables:



Notice that batches 34 and 37 have poor quality attributes

Summary so far

So far we have looked at 3 important ways of using batch data

1. Learning more about the batch systems (all case studies)
2. Troubleshooting problems (DuPont data)
3. Predictive modelling (PLS on the SBR data)

Most (around 90% of batch modelling efforts are related to this).
But we can do more.

Let's look at another important use:

4. Monitoring batch systems
5. Optimizing batch systems by adjusting trajectories

Batch monitoring

Two types of monitoring

1. Off-line, post-batch monitoring

- ▶ Use all the data after the batch is complete: score plots, SPE plots, contribution plots for new data, in the usual way
- ▶ Allows for early release of the batch to the next stage. Don't have to wait for lab results if the batch is multivariately inside the control limits
- ▶ *We have already covered the material for this*
- ▶ Risk: don't just use the SPE and scores at the end of the batch: *it is also **how** you go to the end that matters*

2. On-line monitoring³

- ▶ real-time detection of problems as a new batch progresses
- ▶ many high value batch systems run in the order of weeks
- ▶ save money if we detect and correct these problems before the batch end

³Reference: **Paul Nomikos' PhD thesis**

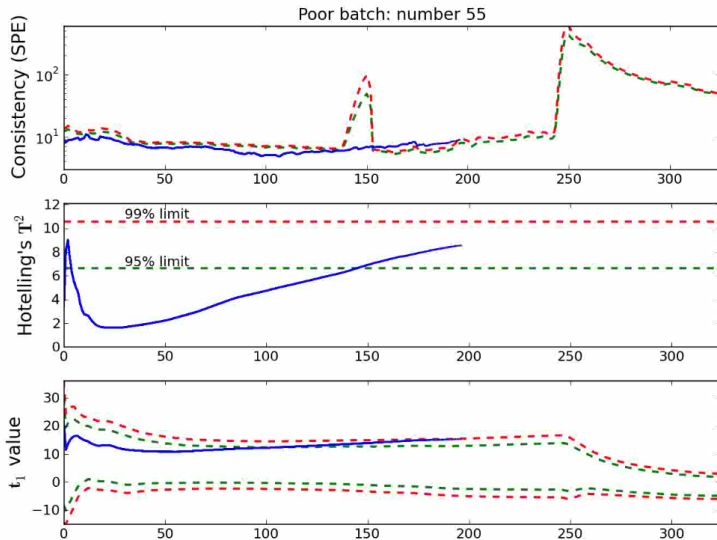
Principle of real-time monitoring (and prediction)

- ▶ While the batch progresses, at time step j , try to get the best estimate of the scores at the **end of the batch**, $\hat{\mathbf{t}}_{j,\text{new}} = \boldsymbol{\tau}_j$
- ▶ $\hat{\mathbf{x}}_{j,\text{new}} = \mathbf{P}_j \boldsymbol{\tau}_j$ ← predicted trajectory at time j
- ▶ $\mathbf{e}_{j,\text{new}} = \mathbf{x}_{j,\text{new}} - \hat{\mathbf{x}}_{j,\text{new}}$ ← only a $K \times 1$ vector
- ▶ $\text{SPE}_{j,\text{new}} = \mathbf{e}_{j,\text{new}}^T \mathbf{e}_{j,\text{new}}$ ← SPE at time j
- ▶ This is called the *instantaneous* SPE
- ▶ $\mathbf{e}_{1:j,\text{new}} = \mathbf{x}_{1:j,\text{new}} - \mathbf{P}_{1:j} \boldsymbol{\tau}_j$ ← a $jK \times 1$ vector
- ▶ SPE calculated using data from start to time j : called the *evolving* SPE
- ▶ Evolving SPE gets closer and closer to final SPE as $j \rightarrow J$
- ▶ For batch PLS, we get a prediction: $\hat{\mathbf{y}}_{j,\text{new}} = \boldsymbol{\tau}_j^T \mathbf{C}$

Our real time monitoring and predictions hinge on the ability to calculate the estimated end-point score, $\hat{\mathbf{t}}_{j,\text{new}} = \boldsymbol{\tau}_j$

Demonstration of batch monitoring

3 monitoring videos: good, poor, and a batch with a problem in the middle



Time-varying monitoring limits

Limits for SPE and the scores vary with time⁴

► SPE limits

- $\text{SPE}_j \sim g \chi^2(h)$ ← follows an approximate χ^2 distribution
- $g = \frac{v}{2m} = \text{premultiplier}$
- $h = \frac{2m^2}{v} = \text{degrees of freedom of } \chi^2(h)$
- $m = \text{mean}(\text{SPE}_j)$
- $v = \text{var}(\text{SPE}_j)$

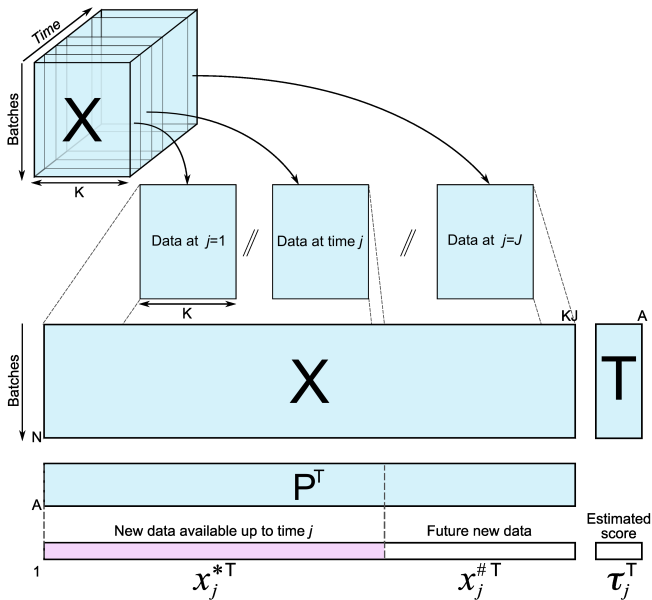
Use SPE values at time step j on all the good batches to estimate g and h

► Score limits

- Assume $t_{a,j}$ to be normally distributed, though a t -distribution is more correct
- Estimate mean and variance at time j from good batches

⁴Derivations in **Nomikos and MacGregor paper**

Real-time monitoring of a new batch



How to handle the missing future data

How to estimate the end-score: $\hat{\mathbf{t}}_{j,\text{new}} = \boldsymbol{\tau}_j$?

1. Fill future value with zeros
 - ▶ implies rest of batch runs at the average trajectory
2. Current deviations approach
 - ▶ mean centered and scaled deviation at time j is copied and pasted forward
 - ▶ implies current deviations persist (MPC assumption)
3. Missing data handling
 - ▶ Use one of the many missing data handling methods for PCA/PLS
 - ▶ score limits tend to have variability at start, but quickly stabilize
 - ▶ single component projection, SCP: poor, but simple choice
 - ▶ projection to model plane, PMP: improves SCP somewhat
 - ▶ conditional mean replacement (CMR) or trimmed score regression (TSR) are better (good)

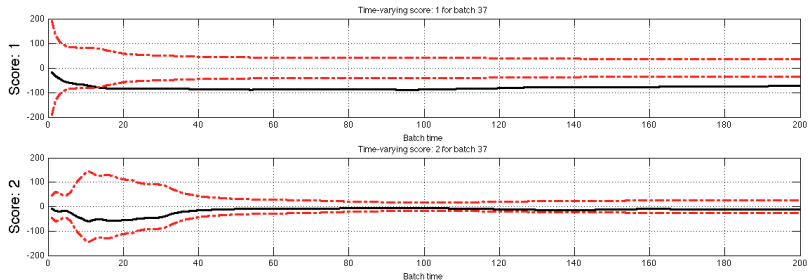
How to handle the missing future data

From a monitoring perspective:

- ▶ doesn't really matter too much which missing data method is used
- ▶ the control limits are a function of the method chosen

More details: [Comparing different missing data approaches](#) for on-line monitoring and trajectory prediction (García-Muñoz *et al.*)

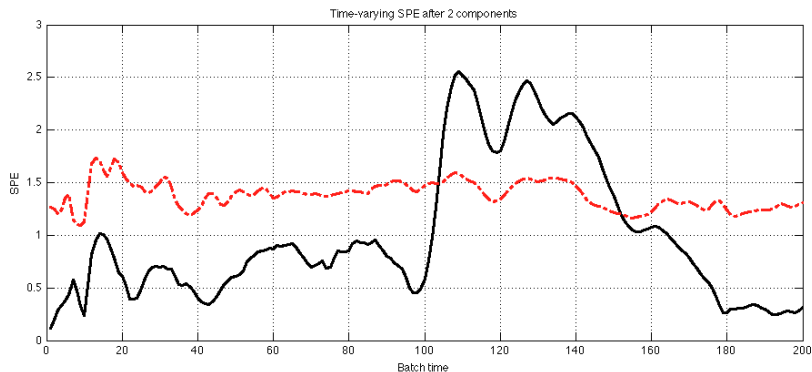
SBR: scores over time for batch 37



- ▶ Highlights *when* the problem occurred: right at the start
 - ▶ Was due to an impurity in the feed: consumed reactant and lowers latex density and conversion
- ▶ SPE was within limits throughout the batch

SBR example: bad batch 34

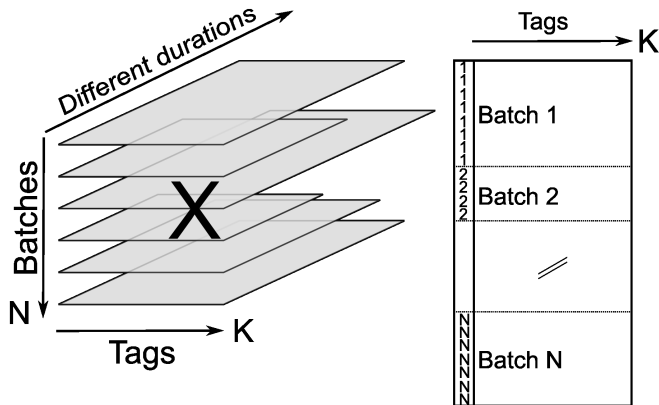
Simulation introduced impurity in feed midway, during the batch



We will use the software to diagnose the contributions

Alignment, and the relationship to unfolding

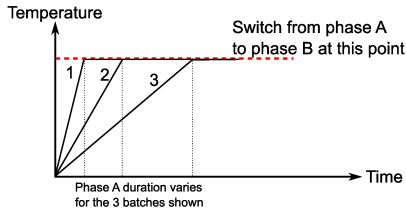
Once you understand *why we unfold* in the batch-wise direction, and also *what is being accomplished* by this unfolding, then **it becomes clear how to align your data**.



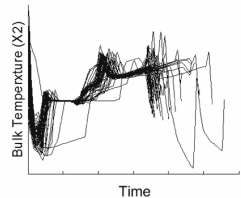
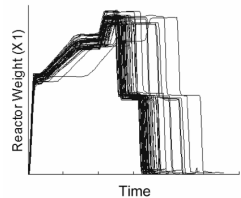
Some automated tools are becoming available. Best results still require case-specific knowledge.

Why uneven duration batches occur

- ▶ Exothermic system with cooling: different batch durations in summer and winter
- ▶ Catalyst and raw material amounts vary
- ▶ Raw material impurities: require longer or shorter times as impurities consume reactants
- ▶ Recipe goes to next phase based on trigger:



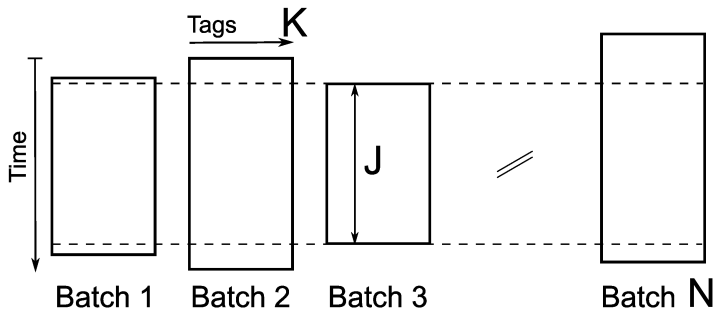
- ▶ We get major alignment problems when batches are operated manually:
 - ▶ “apply agitation until *well mixed*”



Result: similar trajectories of different duration

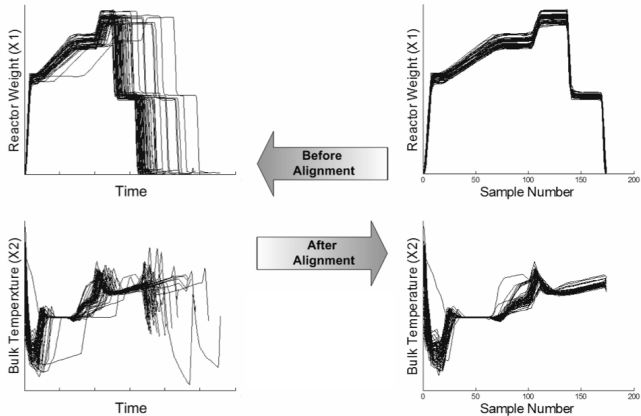
How to align batches

1. *Data trimming*: throw out data points at start or end of each phase to match **shortest duration**.
 - ▶ **Risk**: sometimes most informative data is near start or end
 - ▶ Works well when trajectories are automatically controlled, and batches roughly of equal duration
 - ▶ This was used in the DuPont example.



How to align batches

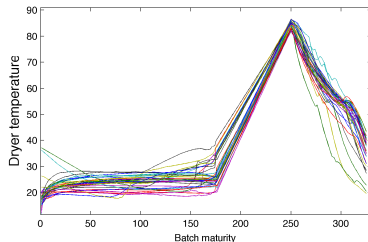
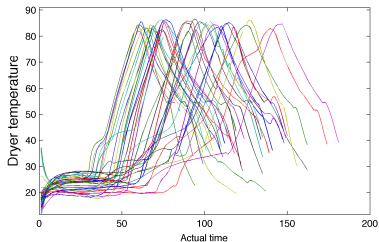
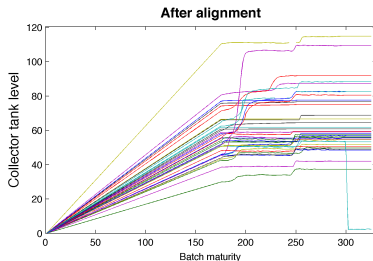
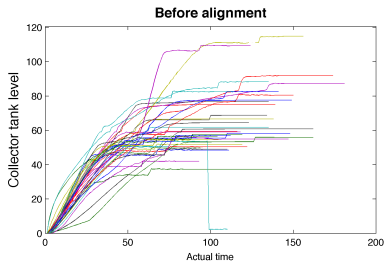
2. *Interpolation*: pick a duration (e.g. the **average duration**), then stretch and shrink other batches by interpolating all trajectories
- ▶ Use interpolation within *each phase*, not over the whole batch



How to align batches

3. *Aligning along an indicator*: any monotonically increasing or decreasing tag within a phase
 - ▶ Adjust all other tags in the phase against this indicator, and interpolate to a chosen number of time points
 - ▶ **Key point**: each batch is adjusted independently of the others
 - ▶ Good results if indicator is related to batch maturity
 - ▶ Create the variable, if necessary: e.g. a cumulative sum
 - ▶ Examples:
 - ▶ temperature ramp
 - ▶ cumulative amount of raw material fed (semi-batch systems)
 - ▶ cumulative energy added to the reactor jacket
 - ▶ a calculated variable, e.g. the total conversion
 - ▶ lance position in injection molding
 - ▶ Can be used later for online, real-time monitoring (e.g. extrapolate temperature slope, use amount of material fed)
 - ▶ Put the alignment information into **Z**: very useful for diagnosis

Example of alignment with an indicator



Other alignment options

Open area of research:

4. *Dynamic time warping*: stretch and shrink data to match a “golden” batch
 - ▶ Kassidas, *et al.*: “Synchronization of batch trajectories using dynamic time warping”, *AIChE Journal*, 1998.
5. *Correlation optimized warping*
 - ▶ Fransson and Folestad: “Real-time alignment of batch process data using COW for on-line process monitoring”, *Chemometrics and Intelligent Laboratory Systems*, 2006.

Where else can batch analysis be applied

Any cyclical process:

- ▶ weather data: annual changes
- ▶ marketing: we know seasonal affects occur
- ▶ manufacturing discrete product: e.g. tool and die applications
- ▶ fermentation processes
 - ▶ beer, wine
- ▶ cycles related to animals and people
 - ▶ sleeping patterns
 - ▶ pregnancy
 - ▶ initial conditions
 - ▶ trajectories: hormones and chemical changes in the body
 - ▶ the entire life cycle: has phases