Chemical Engineering 4H03

PCA Model Fitting Statistics

Jake Nease
McMaster University

Objectives for this Class

- Where we came from…
  - Computing components **P** from NIPALS and EVs
  - Discussion of errors and model residuals

- Now: How do I know if I have enough components?
  - PCA suffers from the same issue as regression: overfitting
  - What are the different types of "errors" of a PCA model?

- How will we do this?
  1. Introduce testing/training sets for PCA models
  2. Introduce modeling statistic: PRESS
  3. Introduce modeling statistic: Q^2
  4. Introduce modeling statistic: Hotelling's T^2

Using a Model on New Data

That's the Goal, Isn't it?

https://img.discogs.com/SVwSVgsJLBsA9QJUghduzF-k8bI=/fit-in/300x300/filters:strip_icc():format(jpeg):mode_rgb():quality(40)/discogs-images/R-2036372-1331401328.jpeg.jpg

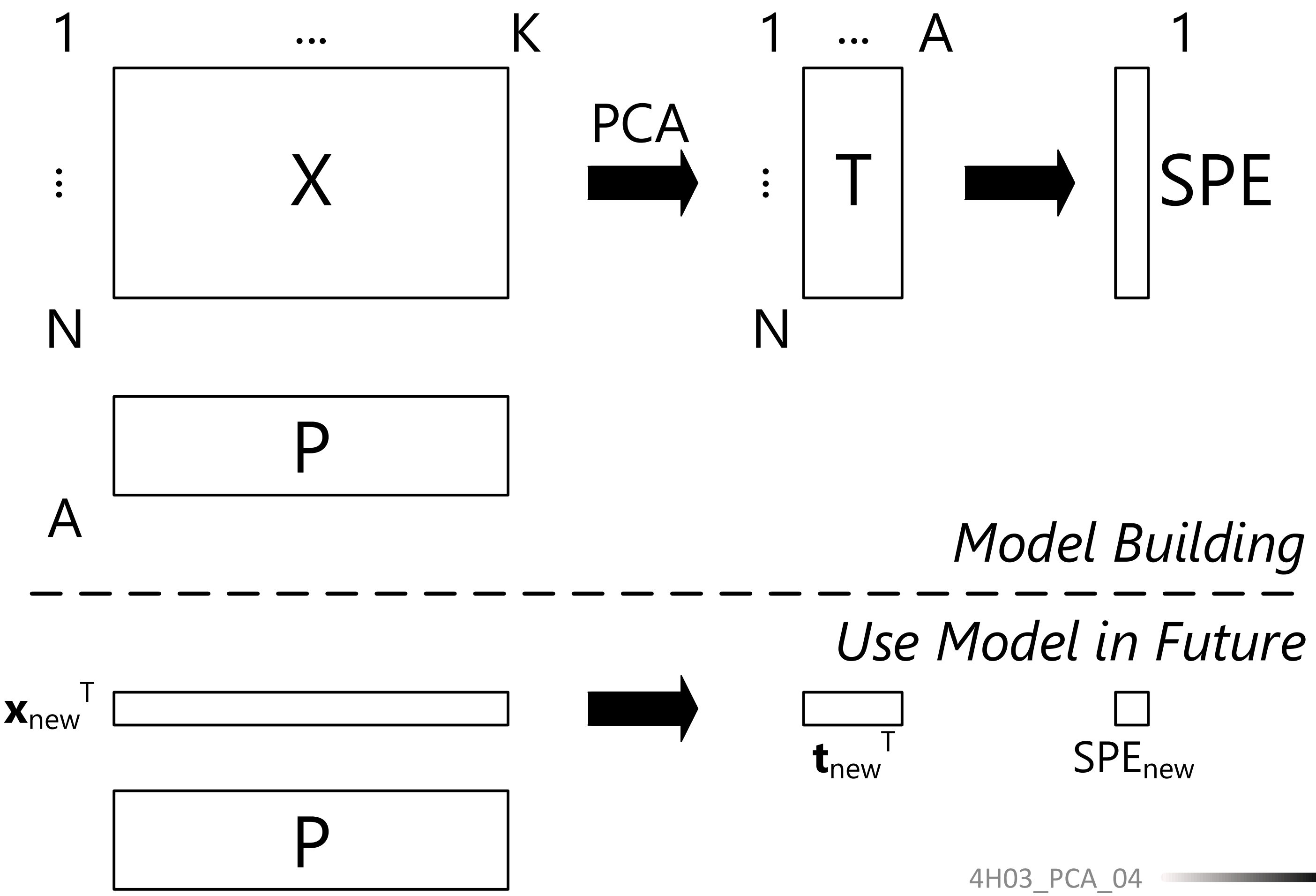
Why Use PCA on NEW Data?

- Several advantages:
  - **Can use modeling statistics to identify outliers (bad data)**
  - Can use PCA model for process monitoring
    - Monitor new scores
    - Monitor SPE values of model predictions
    - Allows use of advanced monitoring statistics
  - Can continue to train model over time
    - Improves model accuracy
  - **Can be used to check model validity**
    - Important application of training and testing data sets!

- The question is… How?

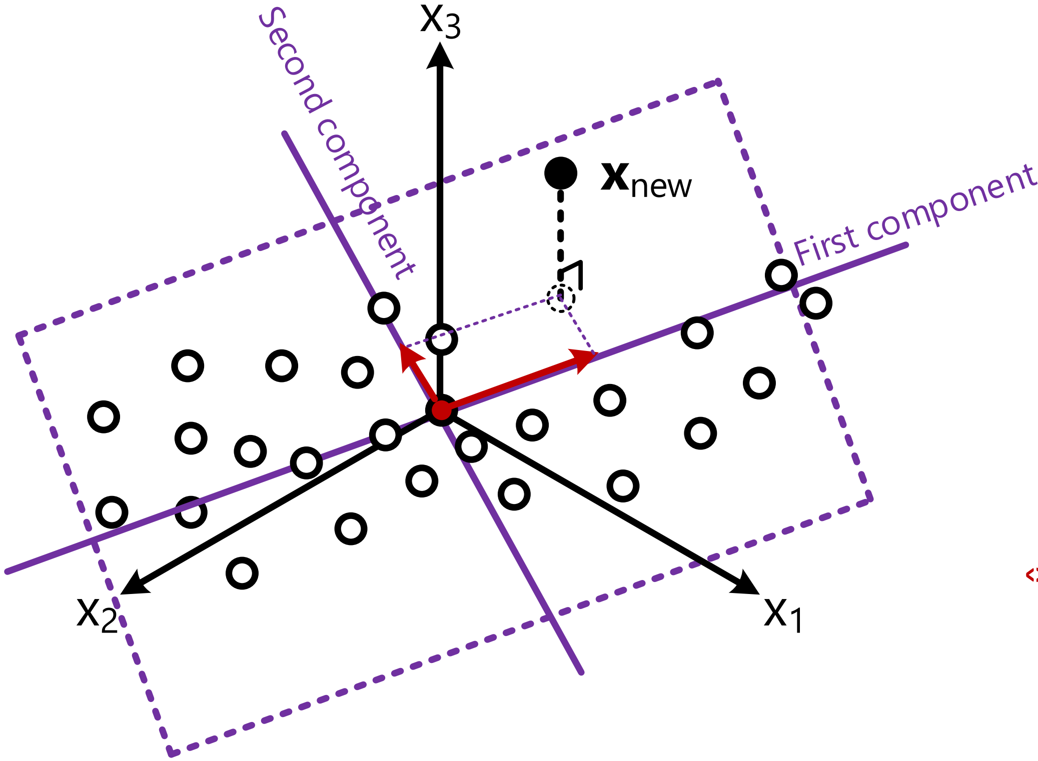
Predicting New Points

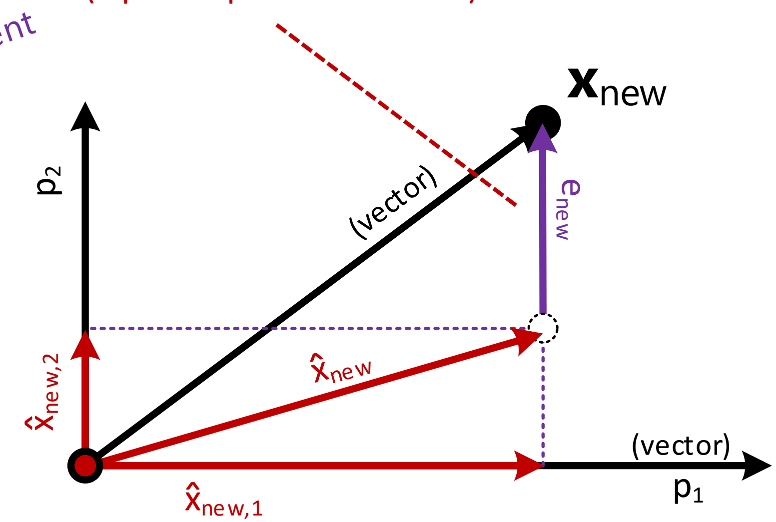
- Recall we could predict $\hat{X}$ to help us identify R^2: $\hat{X} = TP^T$
- Can continue to do this on new data $\boldsymbol{x}_{new}$: $\hat{\boldsymbol{x}}_{new} = \boldsymbol{t}_{new}P^T$



Review: Using a PCA Model

- Consider some new point $\boldsymbol{x}_{new}$ and its relation to the PCA model





$$\boldsymbol{t}_{new}^T =$$

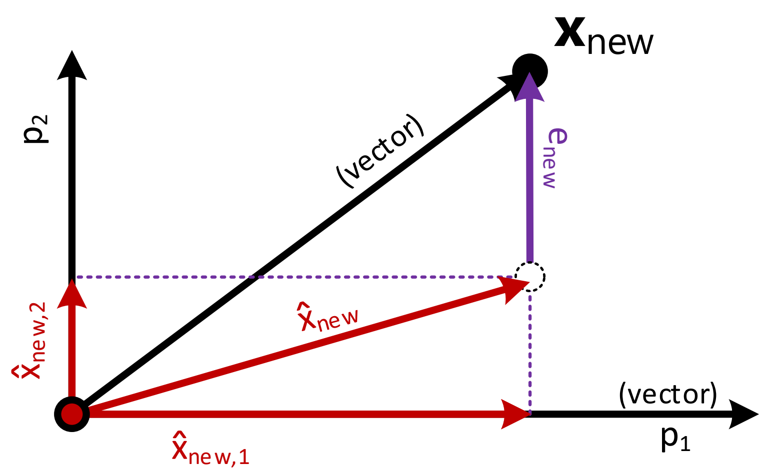
$$\hat{\boldsymbol{x}}_{new}^T =$$

$$\boldsymbol{e}_{new}^T =$$

$$SPE_{new} =$$

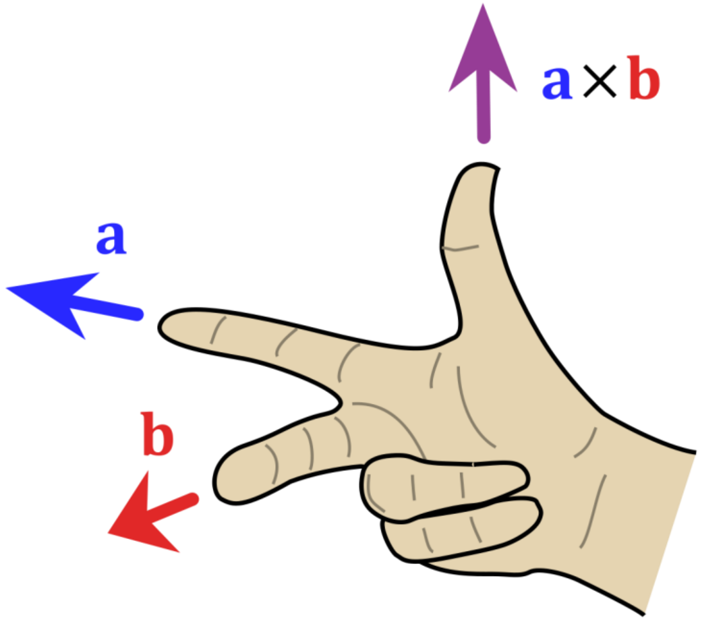
Predicting New Points

- Pretty intuitive, really:
  1. Preprocess data: $\boldsymbol{x}_{new,raw} \rightarrow \boldsymbol{x}_{new}$
  2. Project onto model for scores: $\boldsymbol{t}_{new} = \boldsymbol{x}_{new}^T P$
  3. Use scores to compute prediction: $\widehat{\boldsymbol{x}}_{new} = \boldsymbol{t}_{new}^T P^T$
  4. Compute residuals: $\boldsymbol{e}_{new}^T = \boldsymbol{x}_{new} - \widehat{\boldsymbol{x}}_{new}$
  5. Calculate SPE of point: $SPE_{new} = \boldsymbol{e}_{new}^T\ \boldsymbol{e}_{new}$



Data Cross-Validation

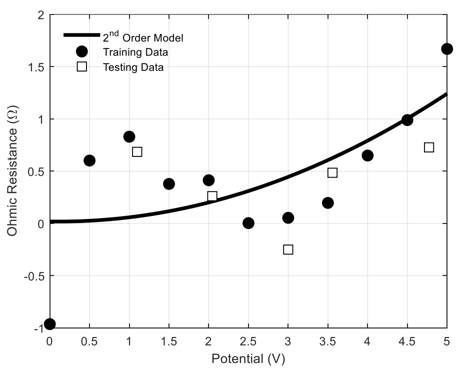
The Cross-Product of Good Modeling



https://upload.wikimedia.org/wikipedia/commons/thumb/d/d2/Right_hand_rule_cross_product.svg/1200px-Right_hand_rule_cross_product.svg.png

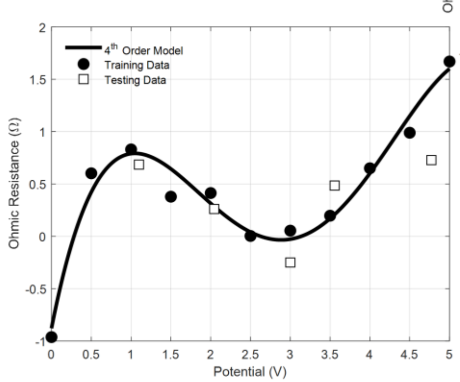
Pitfall of PCA

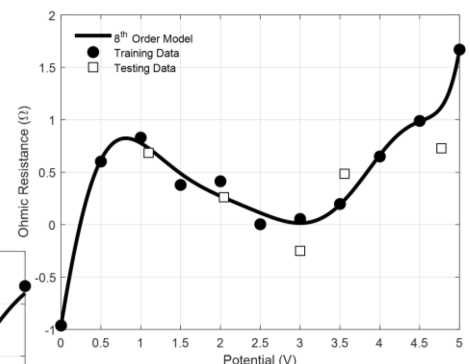
- Much like any regression method, PCA suffers from **overfitting** leading to **model bias**
- Overfitting is the process of **adding complexity** to a model when that complexity is **not supported** by the variance in the data
  - In PCA, this means fitting too many components
  - Akin to fitting only “measurement noise”



R^2 = 0.6

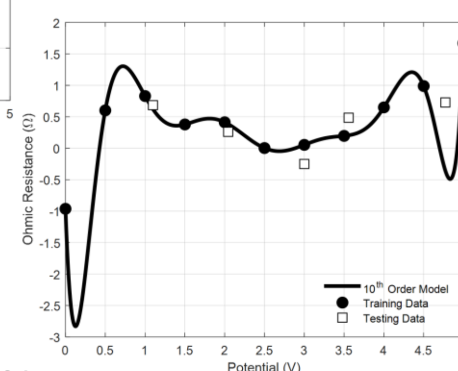
R^2 = 0.85





R^2 = 0.90

R^2 = 1.00…?



Our Strategy

- We desire a PCA model that fits **appropriately**
  - PCA will continue fitting components with diminishing returns on variance explained (see plot)
- What does "appropriately" mean?
  - *We need to know how the model will be used in the future to know if we are overfitting*
- How can we "simulate" future data?
  - You'd best believe it… **Training and Testing data sets**

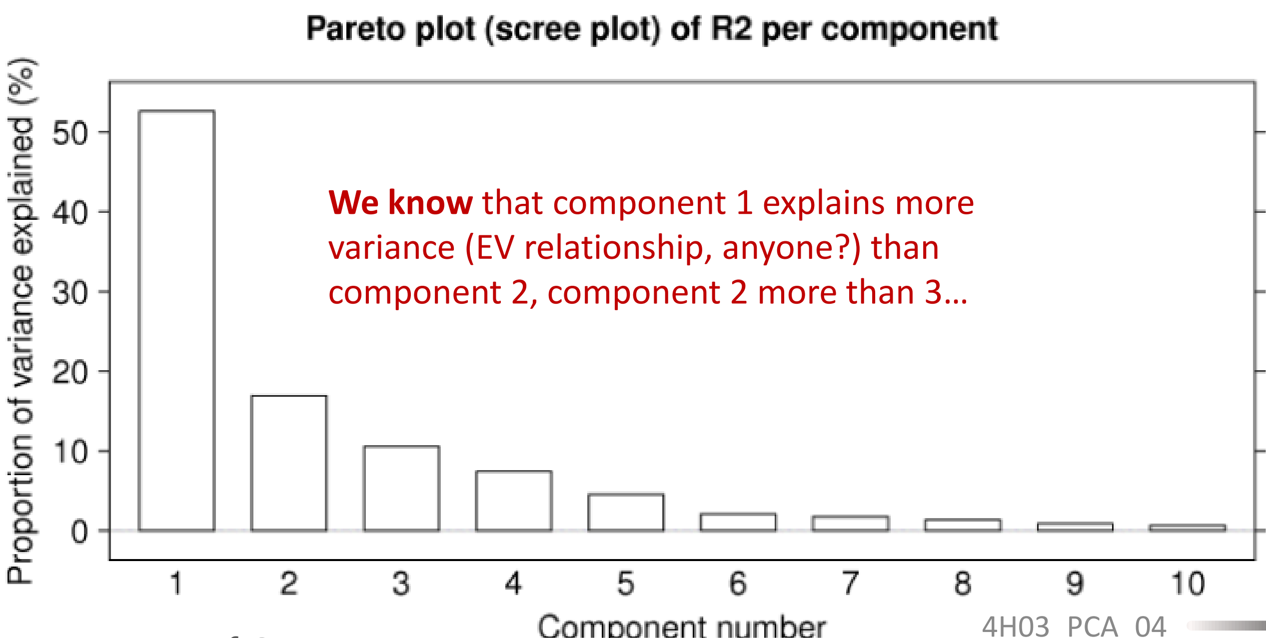


Figure courtesy of ConnectMV

Cross-Validation

- **MISTAKE**: Common to see "fit to a certain value of R^2"
  - This IS ALWAYS POSSIBLE and almost NEVER GOOD
  - Must try to avoid this using clever techniques
  - You can have an $R^2 = 0.90$ be a **terrible model**
  - You can have an $R^2 = 0.90$ be a **terrific model**

- **Cross-Validation:** The act of comparing the predictive performance of a model on a subset of data **not known ahead of time** to the predictive performance of the model using its own **training data**
  - **Objective:** Residuals are "small enough" so that any additional model complexity does not convey additional information

Cross-Validation

- General Strategy:
  1. Keep a testing data set aside
  2. Fit a component to training data
  3. Project testing data onto model
  4. Compute sum-of-squared residuals of testing data
     - This is known as the **PRESS** – prediction error sum-of-squares
  5. Determine if R^2_{TE} and R^2_{TR} are both improving
     - R^2_{TE} in a PCA model is known as Q^2 (so compare R^2 to Q^2)
  6. Repeat from (2)

- Questions
  – What should happen to **PRESS** as A increases?
  – What if we do not have enough data for two sets?

Cross-Validation Calculations

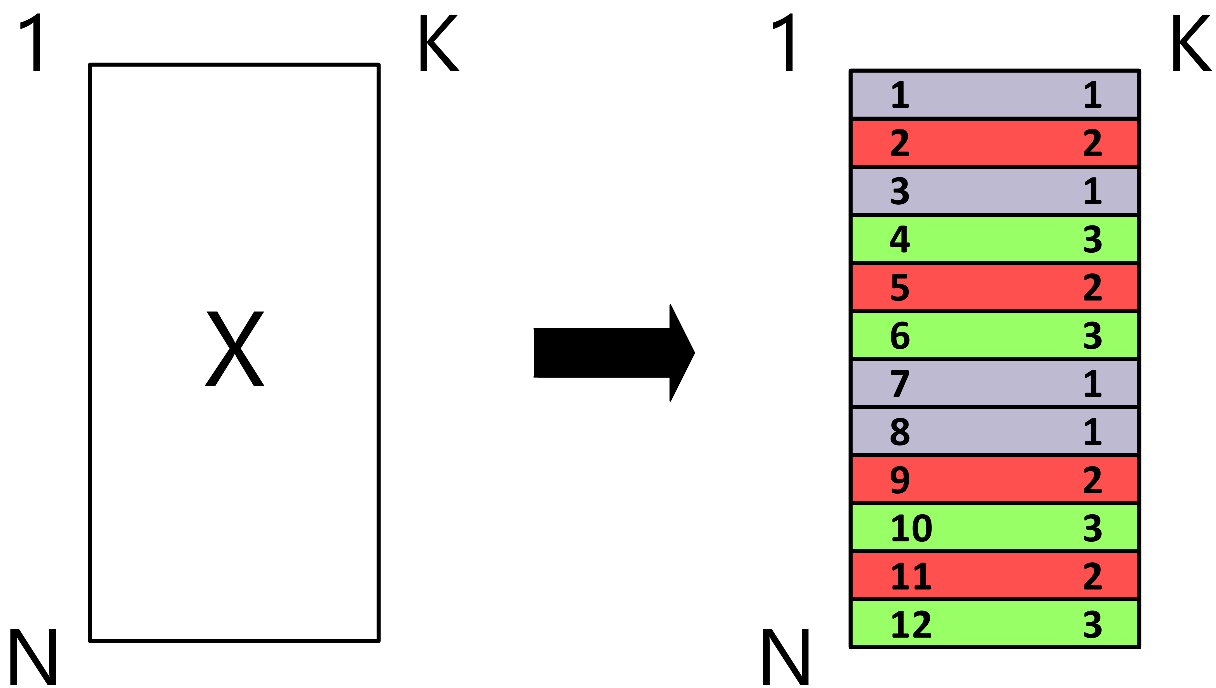
- $\hat{X} = TP^T$
- $X = \hat{X} + E_A$ **Recall** that we have fit **A** components…
- $\mathcal{V}(X) = \mathcal{V}(E_A) + \mathcal{V}(\hat{X})$

- Recall for training data: $R^2 = 1 - \dfrac{\mathcal{V}(E_A)}{\mathcal{V}(X)}$
- DEFINE for testing data: $Q^2 = 1 - \dfrac{\mathcal{V}(E_A\ \text{predicted})}{\mathcal{V}(X)}$

- $\mathcal{V}(E_A\ \text{predicted})$ is known as the **PRESS**

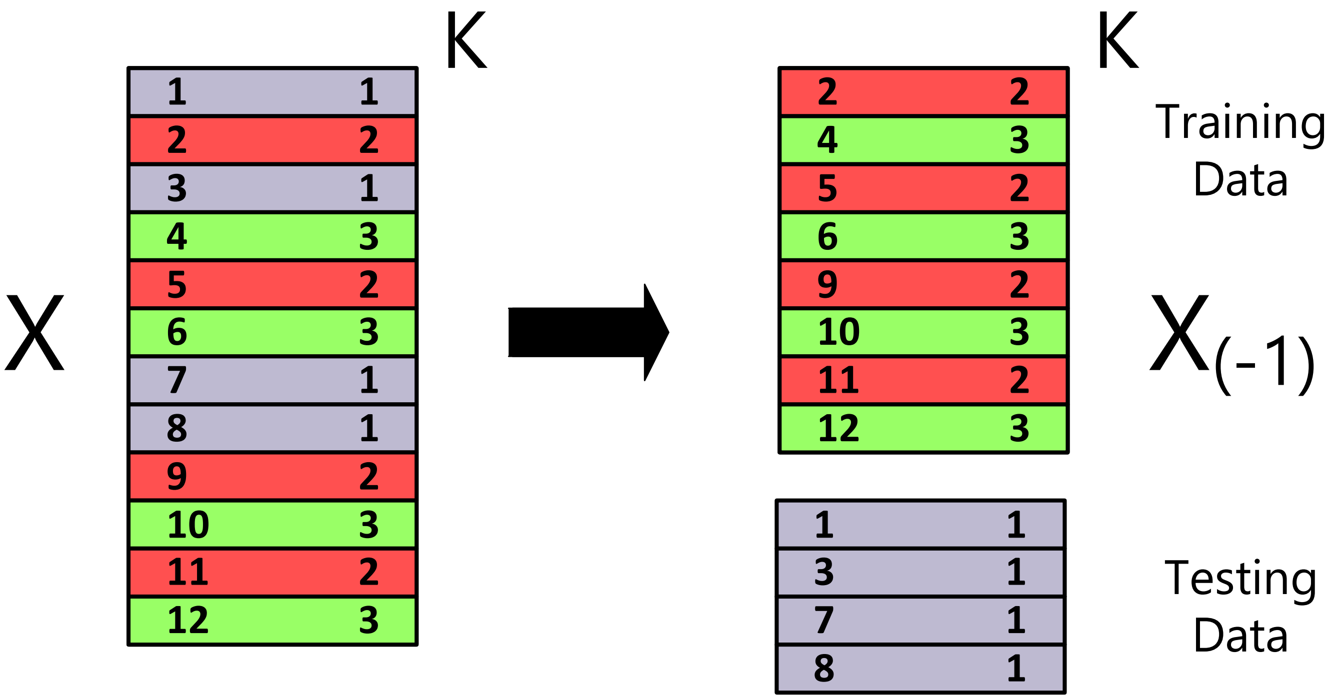
Cross-Validation

- Step 1: Split the rows into **G** groups (3 in this example)
  - Typically **G** is 7 in software packages
  - Can be random or ordered (random in this example)
  - Note that this can depend on time relevance!



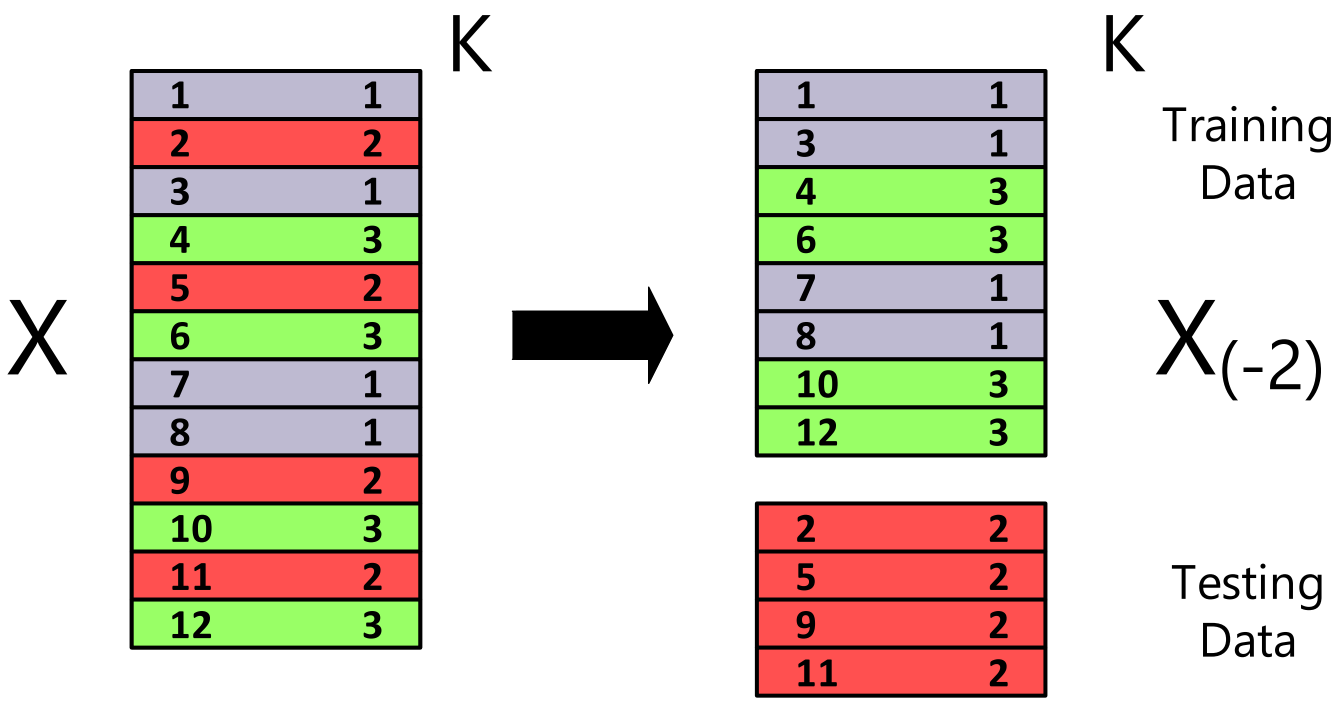
Cross-Validation

- Step 2.1: Fit PCA model
  - Use $\mathbf{X}_{(-1)}$ for fitting
  - Use $\mathbf{X}_{(1)}$ for testing
- Compute $E_{(1)} = X_{(1)} - \hat{X}_{(1)}$



Cross-Validation

- Step 2.2: Fit PCA model
  - Use $\mathbf{X}_{(-2)}$ for fitting
  - Use $\mathbf{X}_{(2)}$ for testing
- Compute $E_{(2)} = X_{(2)} - \hat{X}_{(2)}$



Cross-Validation

- Step 2.3: Fit PCA model
  - Use $\mathbf{X}_{(-3)}$ for fitting
  - Use $\mathbf{X}_{(3)}$ for testing
- Compute $E_{(3)} = X_{(3)} - \hat{X}_{(3)}$

X

	K
1	1
2	2
3	1
4	3
5	2
6	3
7	1
8	1
9	2
10	3
11	2
12	3

$X_{(-3)}$

Training Data

	K
1	1
2	2
3	1
5	2
7	1
8	1
9	2
11	2

Testing Data

4	3
6	3
10	3
12	3

Cross-Validation

- Step 2.G: Fit PCA model
  - Use $\mathbf{X}_{(-G)}$ for fitting
  - Use $\mathbf{X}_{(G)}$ for testing
- Compute $E_{(G)} = X_{(G)} - \hat{X}_{(G)}$

Juuuust pretend I have a nice "Gth" group separated out here…

Q^2 Calculations and Interpretation

- Step 3.1: Calculate **PRESS**
  - $\text{PRESS} = \text{ssq}\left(E_{(1)}\right) + \text{ssq}\left(E_{(2)}\right) + \cdots + \text{ssq}\left(E_{(G)}\right)$
- Step 3.2: Calculate Q^2
  - $Q^2 = 1 - \dfrac{\mathcal{V}(\text{predeicted } E_A)}{\mathcal{V}(X)} \Rightarrow Q^2 = 1 - \dfrac{\text{PRESS}}{\mathcal{V}(X)}$
- Interpretations of Q^2
  - Interpreted the same way as R^2 (higher the better, 1 is best)
  - You should find that $Q^2 \leq R^2$ (unless you are very lucky)
  - If $Q^2 \approx R^2$ for an a^{th} component, the component is **useful**
  - If Q^2 is very small, likely fitting noise
  - Q^2 for an a^{th} components CAN be negative (why?)
  - Q^2_k can be calculated for specific variable k

Cross Validation Summary

- Is there an "autofit" rule?
  - No. BUT there are some heuristics:
    - Keep component if model's Q^2 increases by 1%
    - Keep component if any variable's Q^2_k increases by 5%

- How many components should I use?
  - Depends. Can use cross validation to help guide you
  - Still an open topic in research
  - Does it mean anything? Does it help you solve your objective? If so, keep the component
  - Always fit a few extra components when using software
    - QUESTION: Can you ignore excess components once fit?

Hotelling's T^2

T Time

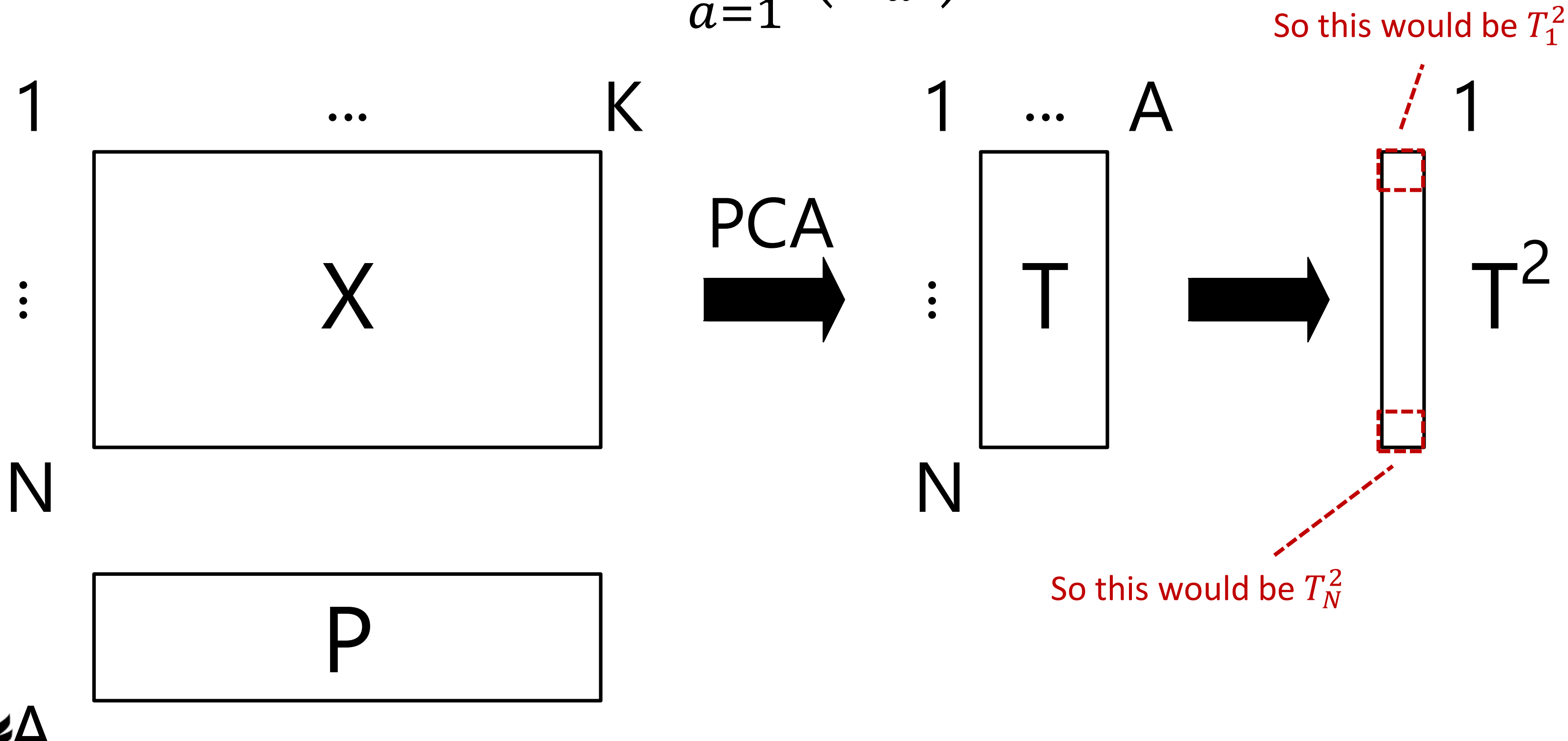
https://i.pinimg.com/originals/70/d5/0a/70d50ac9eb0d2a67018a3c643b5c1cf9.jpg

T^2 In a Nutshell

- After fitting components to **X** we get our scores in **T**
- T_n^2 is the summary of all A components in row n

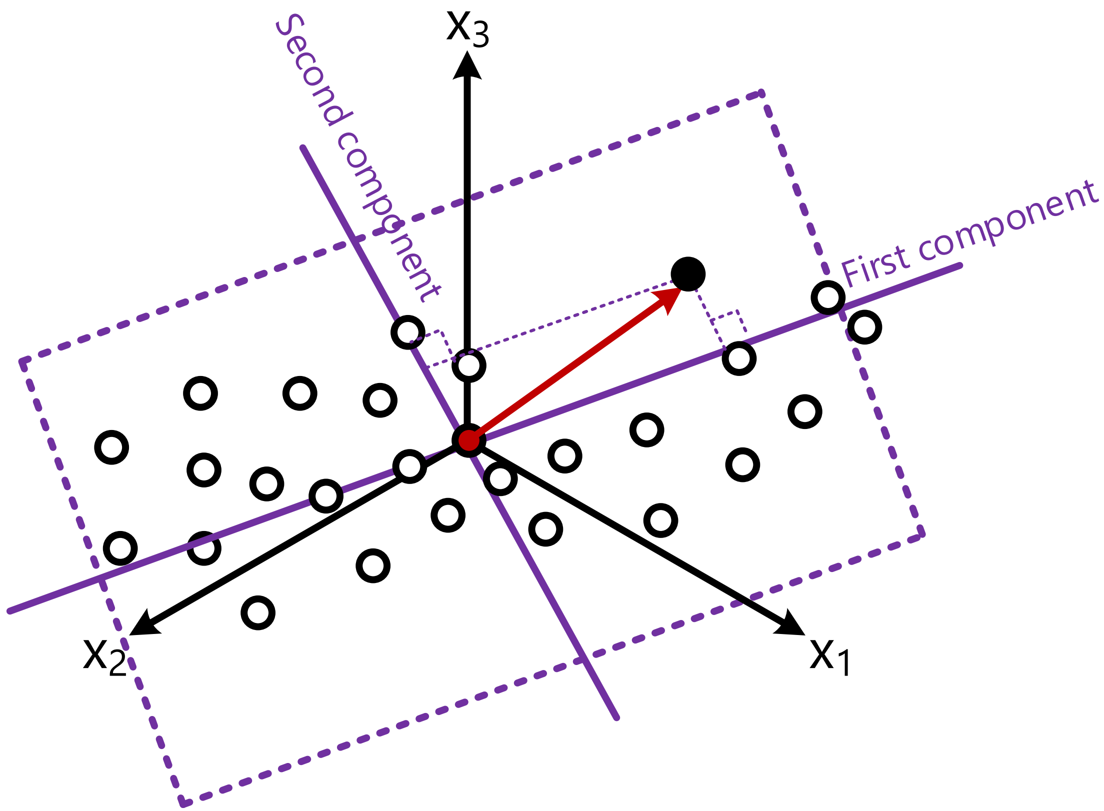
$$T_n^2 = \sum_{a=1}^{A} \left(\frac{t_{n,a}}{s_a} \right)^2$$

s_a is the standard deviation of score column a



T^2 Geometrically

- Represents the **distance** from model center to where a point $\boldsymbol{x}_n$ is projected on the model plane
  - NOT to be confused with distance OFF the plane, AKA SPE



T^2 Properties

$$T_n^2 = \sum_{a=1}^{A} \left(\frac{t_{n,a}}{s_a} \right)^2$$

- Recall $s_1 > s_2 > s_3$ (remember Eigenvalues Decomp?)
  - DISCUSSION: What does this mean WRT where the "distance" components come from when deviating from origin?
  - $T_n^2 > 0$
- Can be plotted as a time series
  - Will flag any point that is "on model plane" but is an extreme case (AKA extremely oily or extremely hard)
  - Useful if rows in data have a meaning
    - Samples taken over time
    - Performance of a plant or piece of equipment during maintenance
- T^2 follows an F-Distribution
  - Allows for confidence intervals (e.g. 95%) to be made
  - Up to you to explore in an assignment

T^2 Properties

$$T_n^2 = \sum_{a=1}^{A} \left(\frac{t_{n,a}}{s_a} \right)^2$$

- In general, for A components and N observations, the α confidence limit:

$$T_{A,\alpha}^2 = \frac{(N-1)(N+1)A}{N(N-A)} F_\alpha(A, N-A)$$

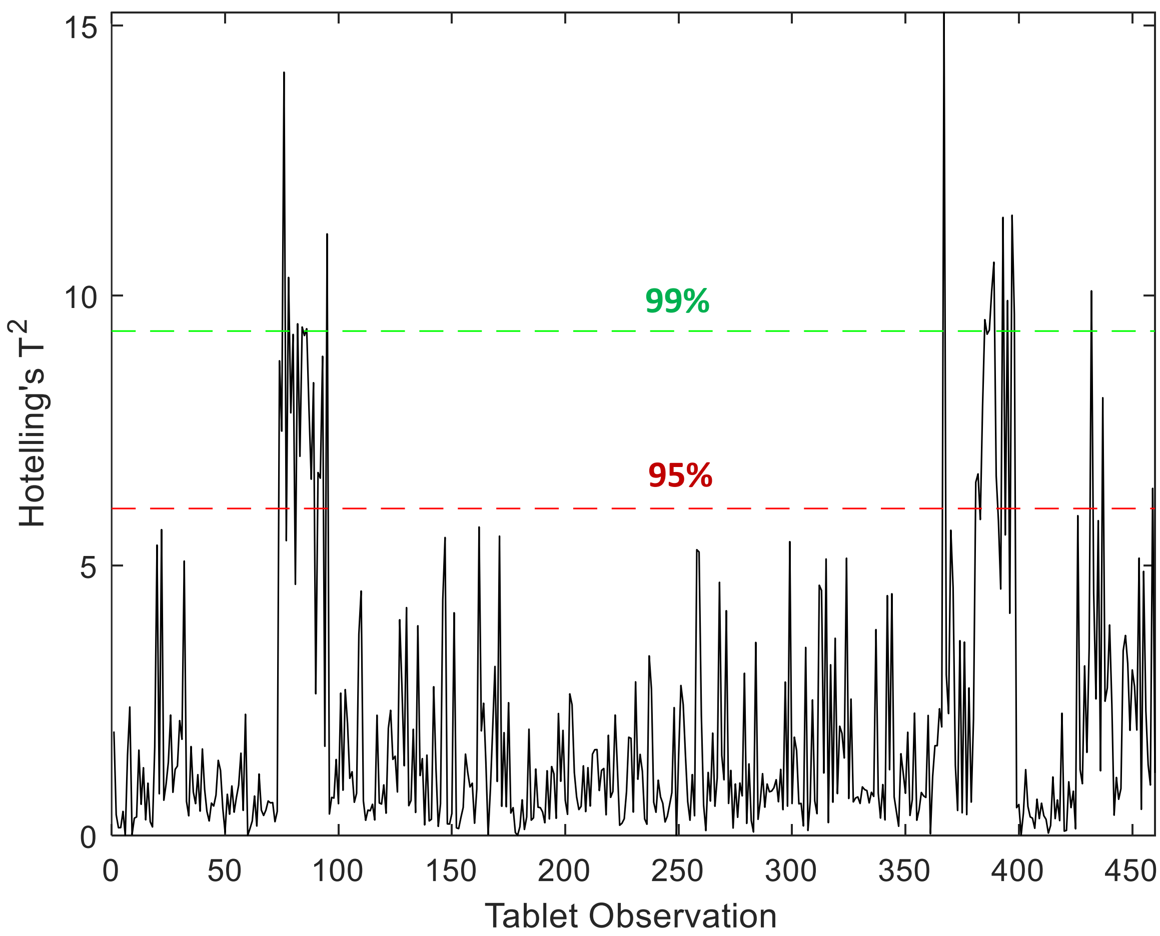
F distribution with numerator A and denominator (N-A), with tail length α

- Special
  - Demo in MATLABnote: Can calculate $F_\alpha(\cdot)$ using `finv` in MATLAB or Python

T^2 for Spectral Data

$$T_n^2 = \sum_{a=1}^{A} \left(\frac{t_{n,a}}{s_a} \right)^2$$

- So now we can see tablets that were "strange"



Final Remarks

- PCA models should not be over-fit
  - We have a good tool to use now: PRESS → Q^2
- SPE is good at flagging observations off model plane
  - Shows a data point is not conforming to known correlation
  - ALSO useful for identifying outliers in training data set
- Hotelling's T^2 is an indication of distance *on* model plane
  - Useful for identifying data that "follow rules" but are extreme cases
  - VERY useful for identifying outliers in training phase (why?)
- Next up
  - Extension of PCA to predicting outcomes via PLS
  - And finally, on to machine learning!