Dimensionality Reduction
Part 1: Matrix Factorization Techniques

Gabrielle M. Schroeder

CNNP Journal Club

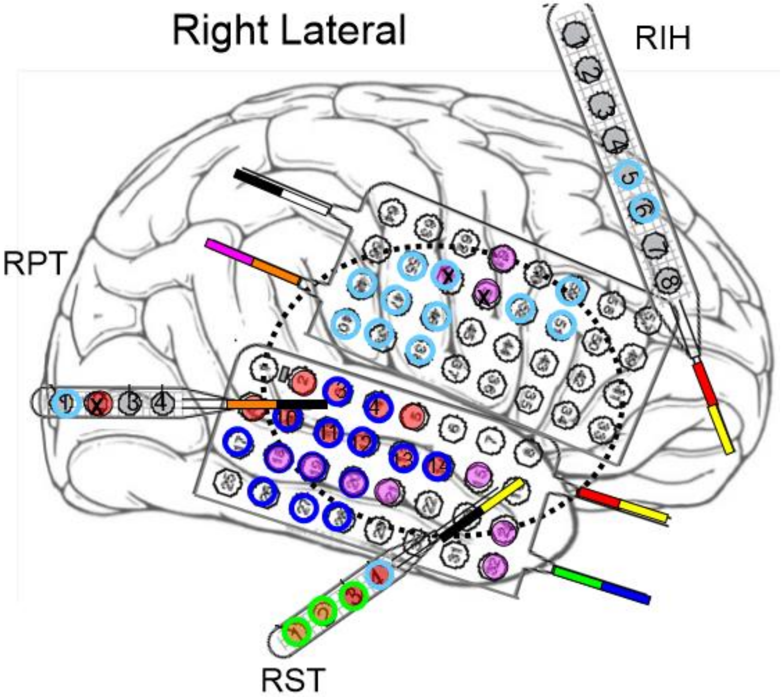
29 January 2018

Contact: g.m.schroeder1@ncl.ac.uk

Why use dimensionality reduction techniques?

Dimensionality reduction: reducing the number of variables in a data set

- Feature selection: find subset of the variables
- **Feature extraction:** transform the data into a lower-dimensional space



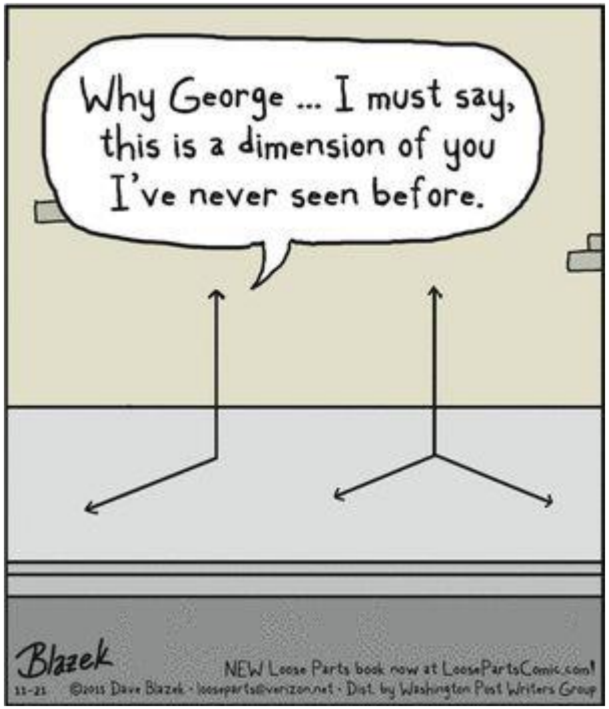
Uses:

- Visualize data
- Find structure in data
- Remove noise
- Avoid "curse of dimensionality"

https://en.wikipedia.org/wiki/Dimensionality_reduction
https://en.wikipedia.org/wiki/Curse_of_dimensionality

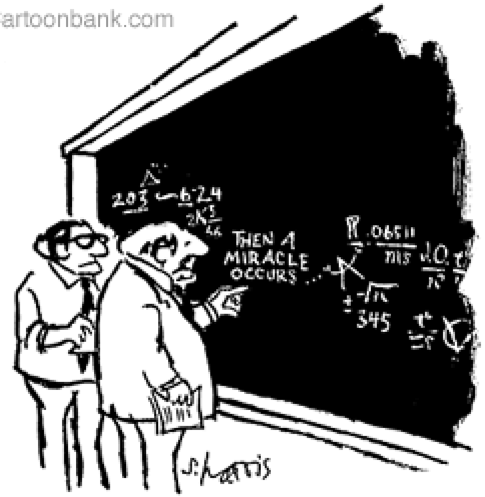
Dimensionality reduction

- **Part 1: Matrix factorization techniques**
  - Principal components analysis (PCA)
  - Non-negative matrix factorization (NMF or NNMF)
  - Other methods: Singular value decomposition (SVD), linear discriminant analysis (LDA), independent components analysis (ICA), tensor decomposition
- **Part 2: Embedding algorithms/manifold learning:**
  - Multidimensional scaling (MDS)
  - t-SNE


Why George ... I must say, this is a dimension of you I've never seen before.
Blazek
NEW Loose Parts book now at LoosePartsComic.com!
11-21 ©2011 Dave Blazek · looseparts@verizon.net · Dist. by Washington Post Writers Group

Relevant references will be on the bottom of each slide

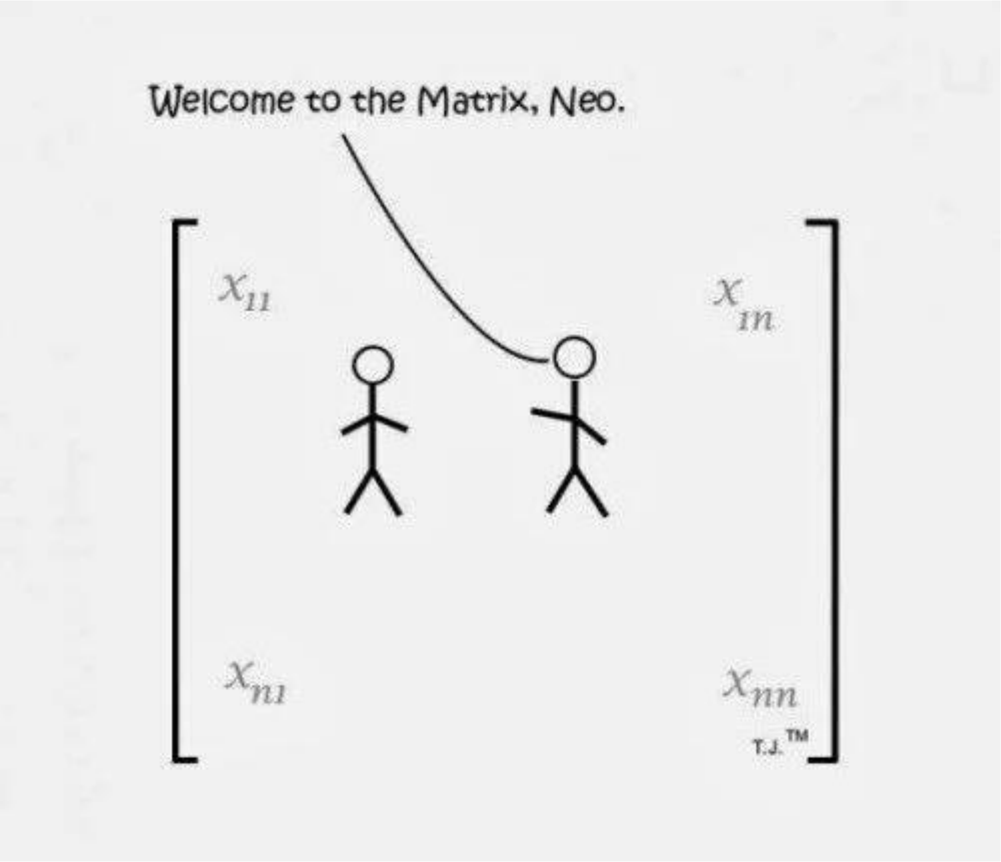
© Cartoonbank.com



"I think you should be more explicit here in step two."

A brief linear algebra review…

Matrix Multiplication



References:

- https://web.stanford.edu/class/nbio228-01/ (Linear Algebra Lecture 1)

Multiplication:

Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

Multiplication:

Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = x_1 y_1$$

Multiplication:
Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = x_1 y_1 + x_2 y_2$$

Multiplication:

Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N$$

Multiplication:
Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

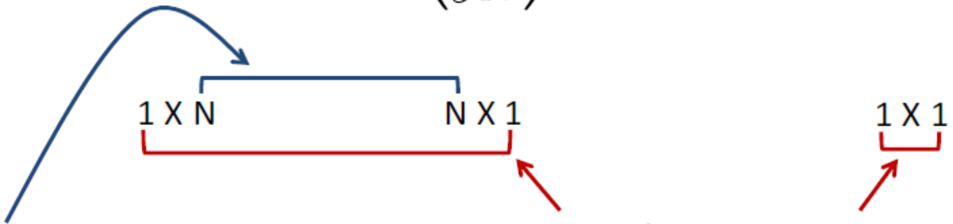
$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N$$

$$= \sum_{i=1}^{N} x_i y_i$$

Multiplication:
Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N$$



1 X N

N X 1

1 X 1

- MATLAB: ‘inner matrix dimensions must agree’

Outer dimensions give size of resulting matrix

Matrix times Matrix: **by inner products**

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{i1} & A_{i2} & \cdots & A_{iP} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1j} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2j} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{Pj} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & C_{ij} & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

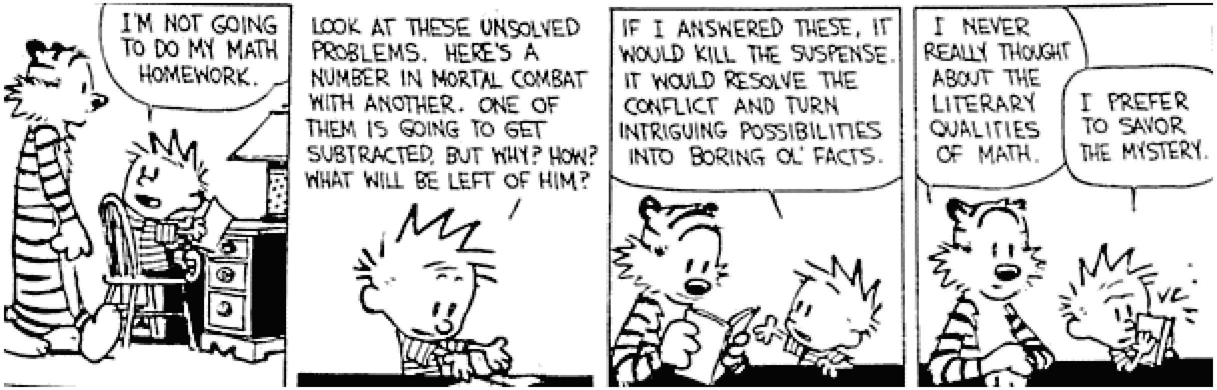
- C_{ij} is the inner product of the ith row of A with the jth column of B

Matrix multiplication: Exercise

Find the product AB for the following matrices:

$$A = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 3 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 3 \\ -2 & -1 \\ 0 & 4 \end{bmatrix}$$


I'M NOT GOING TO DO MY MATH HOMEWORK.
LOOK AT THESE UNSOLVED PROBLEMS. HERE'S A NUMBER IN MORTAL COMBAT WITH ANOTHER. ONE OF THEM IS GOING TO GET SUBTRACTED, BUT WHY? HOW? WHAT WILL BE LEFT OF HIM?
IF I ANSWERED THESE, IT WOULD KILL THE SUSPENSE. IT WOULD RESOLVE THE CONFLICT AND TURN INTRIGUING POSSIBILITIES INTO BORING OL' FACTS.
I NEVER REALLY THOUGHT ABOUT THE LITERARY QUALITIES OF MATH.
I PREFER TO SAVOR THE MYSTERY.

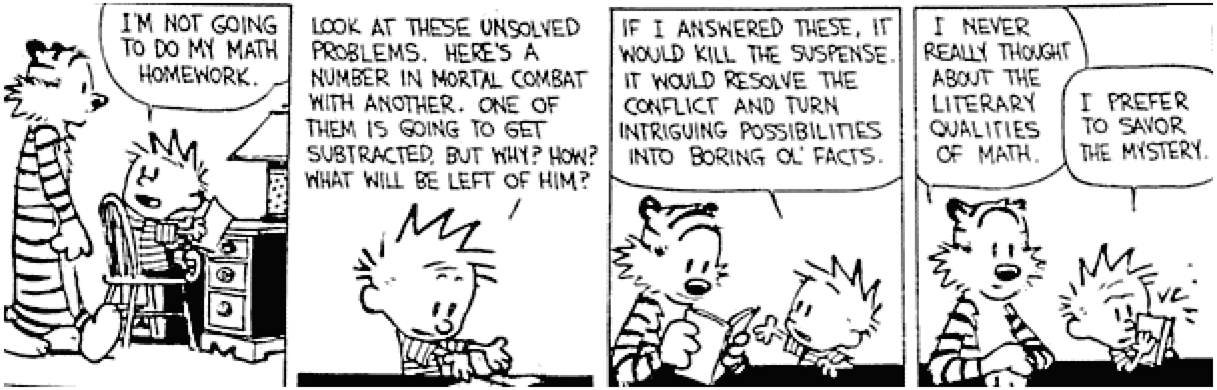
Matrix multiplication: Exercise

Find the product AB for the following matrices:

$$A = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 3 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 3 \\ -2 & -1 \\ 0 & 4 \end{bmatrix}$$

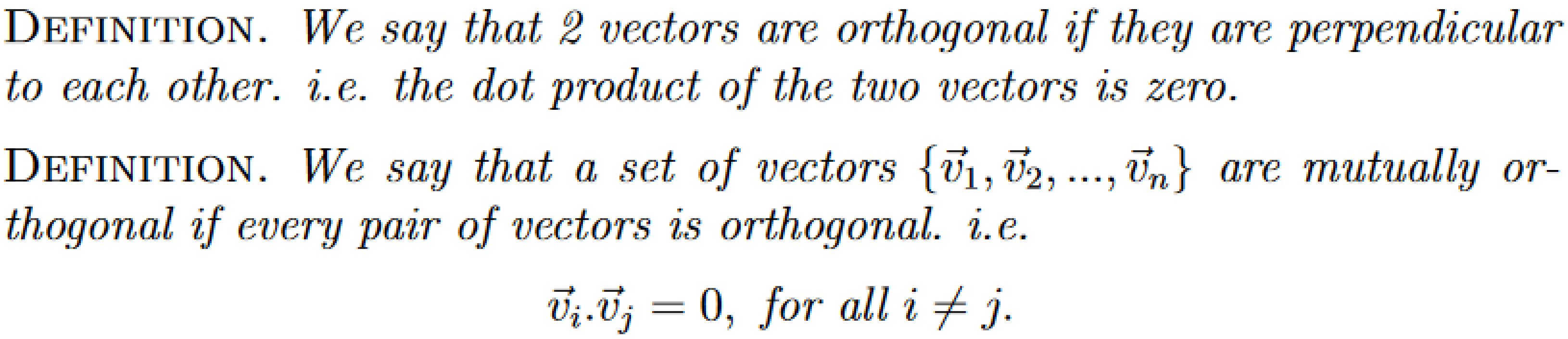
$$AB = \begin{bmatrix} 0 & -5 \\ -6 & -7 \end{bmatrix}$$


I'M NOT GOING TO DO MY MATH HOMEWORK.
LOOK AT THESE UNSOLVED PROBLEMS. HERE'S A NUMBER IN MORTAL COMBAT WITH ANOTHER. ONE OF THEM IS GOING TO GET SUBTRACTED. BUT WHY? HOW? WHAT WILL BE LEFT OF HIM?
IF I ANSWERED THESE, IT WOULD KILL THE SUSPENSE. IT WOULD RESOLVE THE CONFLICT AND TURN INTRIGUING POSSIBILITIES INTO BORING OL' FACTS.
I NEVER REALLY THOUGHT ABOUT THE LITERARY QUALITIES OF MATH.
I PREFER TO SAVOR THE MYSTERY.

References

- http://www.purplemath.com/modules/mtrxmult.htm

Orthogonality



Definition. *We say that 2 vectors are orthogonal if they are perpendicular to each other. i.e. the dot product of the two vectors is zero.*

Definition. *We say that a set of vectors* $\{\vec{v}_1, \vec{v}_2, ..., \vec{v}_n\}$ *are mutually orthogonal if every pair of vectors is orthogonal. i.e.*

$$\vec{v}_i.\vec{v}_j = 0, \textit{ for all } i \neq j.$$

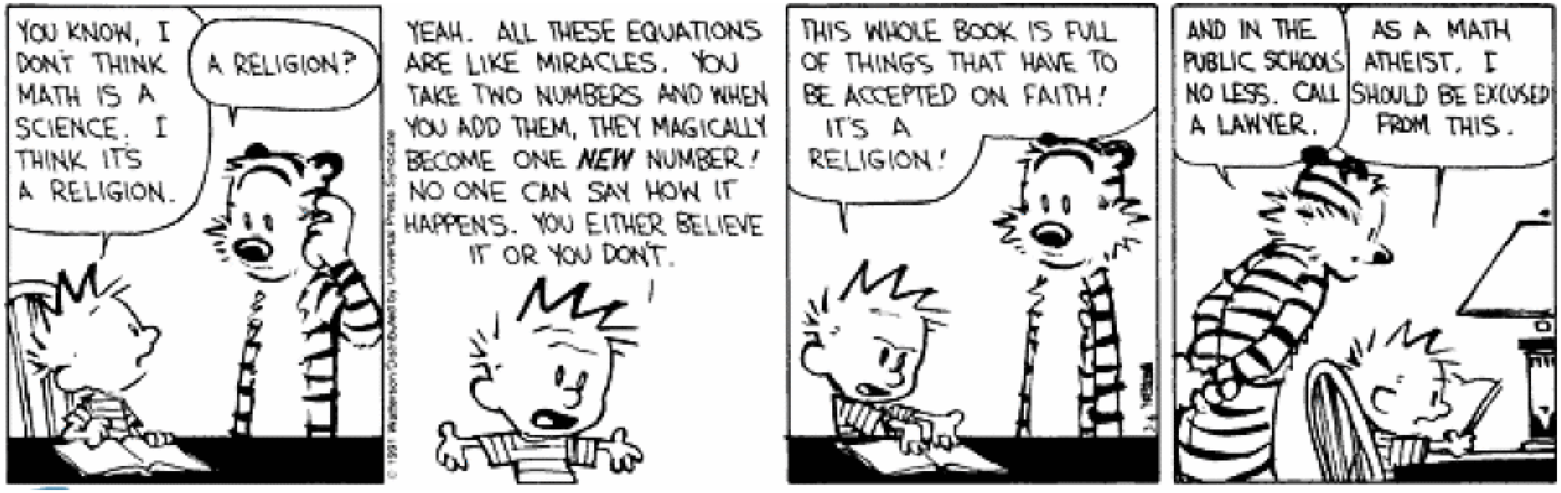
References

- http://www.ucl.ac.uk/~ucahmdl/LessonPlans/Lesson10.pdf

Orthogonality: Exercise

- Are these vectors orthogonal?

$$\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ \sqrt{2} \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -\sqrt{2} \\ 1 \end{pmatrix}$$


YOU KNOW, I DON'T THINK MATH IS A SCIENCE. I THINK IT'S A RELIGION.
A RELIGION?
YEAH. ALL THESE EQUATIONS ARE LIKE MIRACLES. YOU TAKE TWO NUMBERS AND WHEN YOU ADD THEM, THEY MAGICALLY BECOME ONE NEW NUMBER! NO ONE CAN SAY HOW IT HAPPENS. YOU EITHER BELIEVE IT OR YOU DON'T.
THIS WHOLE BOOK IS FULL OF THINGS THAT HAVE TO BE ACCEPTED ON FAITH! IT'S A RELIGION!
AND IN THE PUBLIC SCHOOLS NO LESS. CALL A LAWYER.
AS A MATH ATHEIST, I SHOULD BE EXCUSED FROM THIS.

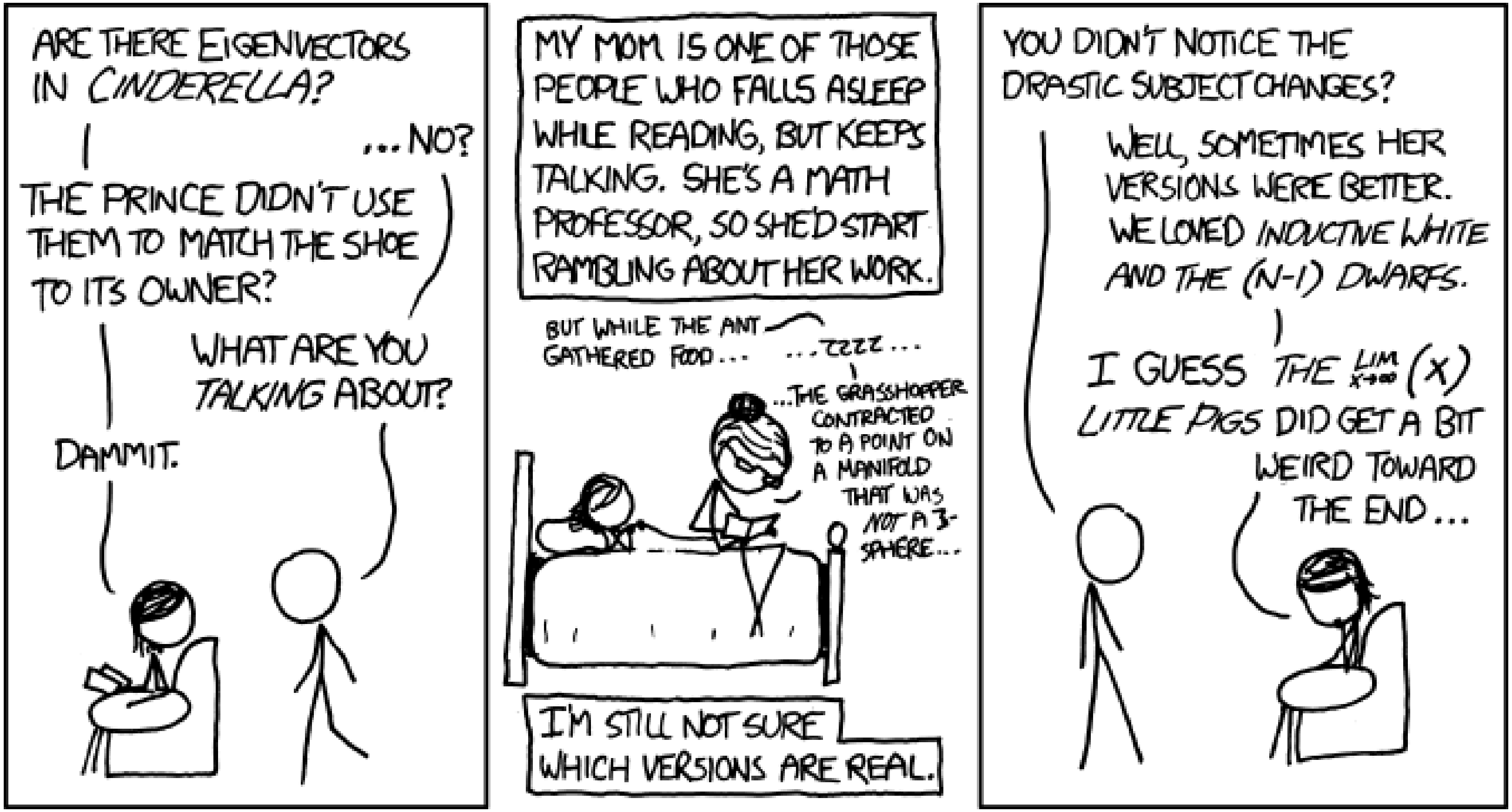
Orthogonality: Exercise

- Are these vectors orthogonal?

$$\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ \sqrt{2} \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -\sqrt{2} \\ 1 \end{pmatrix}$$

$$\begin{aligned} (1,0,-1).(1,\sqrt{2},1) &= 0 \\ (1,0,-1).(1,-\sqrt{2},1) &= 0 \\ (1,\sqrt{2},1).(1,-\sqrt{2},1) &= 0 \end{aligned}$$

Eigenvectors


ARE THERE EIGENVECTORS IN *CINDERELLA?*
...NO?
THE PRINCE DIDN'T USE THEM TO MATCH THE SHOE TO ITS OWNER?
WHAT ARE YOU *TALKING* ABOUT?
DAMMIT.
MY MOM IS ONE OF THOSE PEOPLE WHO FALLS ASLEEP WHILE READING, BUT KEEPS TALKING. SHE'S A MATH PROFESSOR, SO SHE'D START RAMBLING ABOUT HER WORK.
BUT WHILE THE ANT GATHERED FOOD...
...ZZZZ...
...THE GRASSHOPPER CONTRACTED TO A POINT ON A MANIFOLD THAT WAS *NOT* A 3-SPHERE...
I'M STILL NOT SURE WHICH VERSIONS ARE REAL.
YOU DIDN'T NOTICE THE DRASTIC SUBJECT CHANGES?
WELL, SOMETIMES HER VERSIONS WERE BETTER. WE LOVED *INDUCTIVE WHITE AND THE (N-1) DWARFS.*
I GUESS *THE* $\lim_{x\to\infty}(x)$ *LITTLE PIGS* DID GET A BIT WEIRD TOWARD THE END...

Eigenvectors

$$\begin{bmatrix} 0 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} =$$

Eigenvectors

$$\begin{bmatrix} 0 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 3 \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Eigenvectors

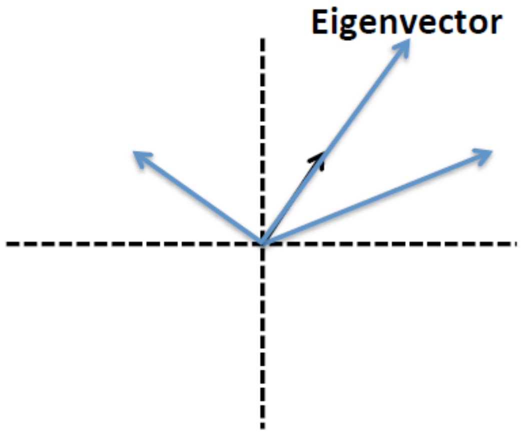
$$\begin{bmatrix} 0 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 3 \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\overleftrightarrow{W}\,\underset{\underset{\textbf{Eigenvector}}{\uparrow}}{\overrightarrow{v}} = \underset{\underset{\textbf{Eigenvalue}}{\uparrow}}{\lambda}\,\overrightarrow{v}$$

Eigenvectors

$$\begin{bmatrix} 0 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 3 \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\overleftrightarrow{W}\underset{\substack{\uparrow \\ \textbf{Eigenvector}}}{\overrightarrow{v}} = \underset{\substack{\uparrow \\ \textbf{Eigenvalue}}}{\lambda}\overrightarrow{v}$$



For an eigenvector, multiplication by the matrix is the same as multiplying the vector by a scalar. This scalar is called an eigenvalue.

How many eigenvectors can a matrix have???

→ An *n* x *n* matrix (the only type we will consider here) can have <u>up to</u> *n* distinct eigenvalues and *n* eigenvectors

→ The set of eigenvectors, with distinct eigenvalues, are linearly independent. For symmetric matrices: they are orthogonal (their dot product is 0).

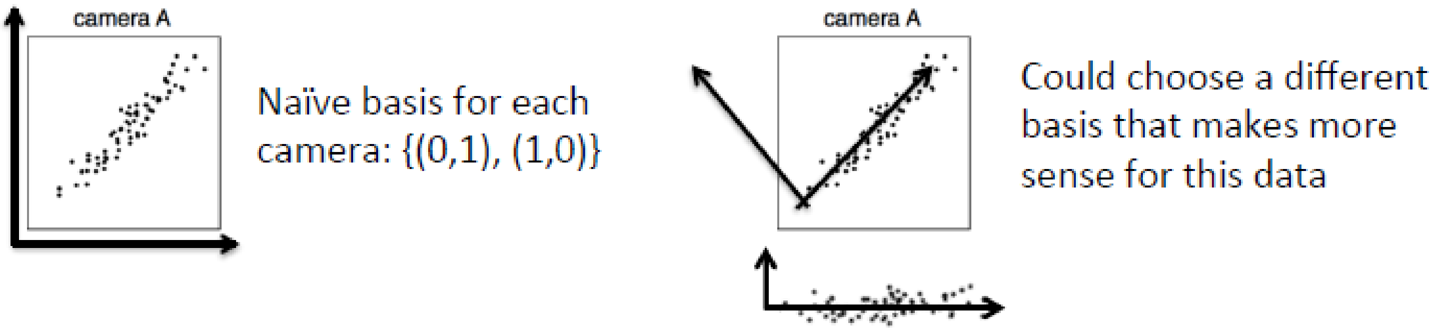
References:

- https://web.stanford.edu/class/nbio228-01/ (Linear Algebra Lecture 2)

Principal Components Analysis

PCA

- **Goal:** Given a set of data with n observations with p variables (p dimensions), reduce the dimensionality of the data while **maintaining as much of the variability** in the data as possible.



Naïve basis for each camera: {(0,1), (1,0)}

Could choose a different basis that makes more sense for this data

References:

- https://web.stanford.edu/class/nbio228-01/ (Linear Algebra Lecture 2)
- **Everything you did and didn’t know about PCA:** http://alexhwilliams.info/itsneuronalblog/2016/03/27/pca/

PCA

How:

Find linear combinations (= **principal components**) of the original p variables that capture directions of maximum variation in the data.

Express the data in terms of this new basis of principal components.

PCA

How:

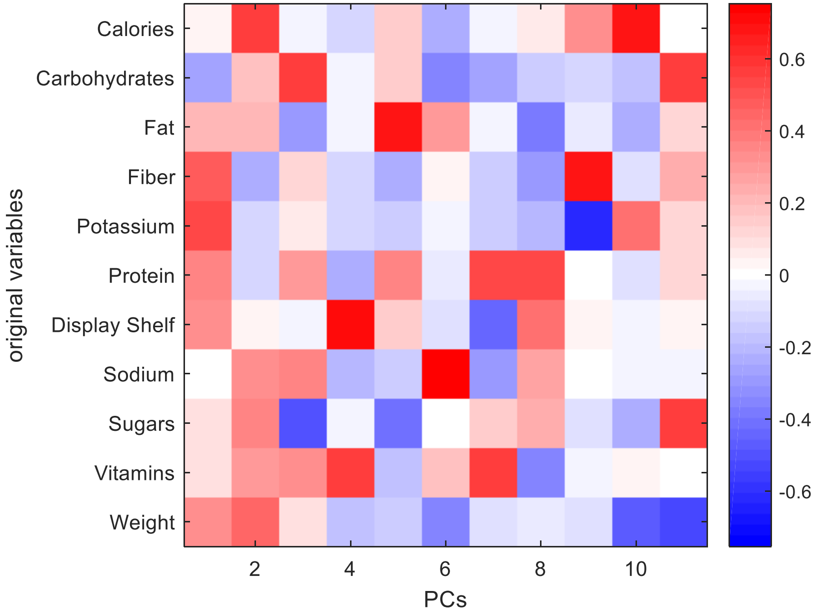
Find linear combinations (= **principal components**) of the original p variables that capture directions of maximum variation in the data.

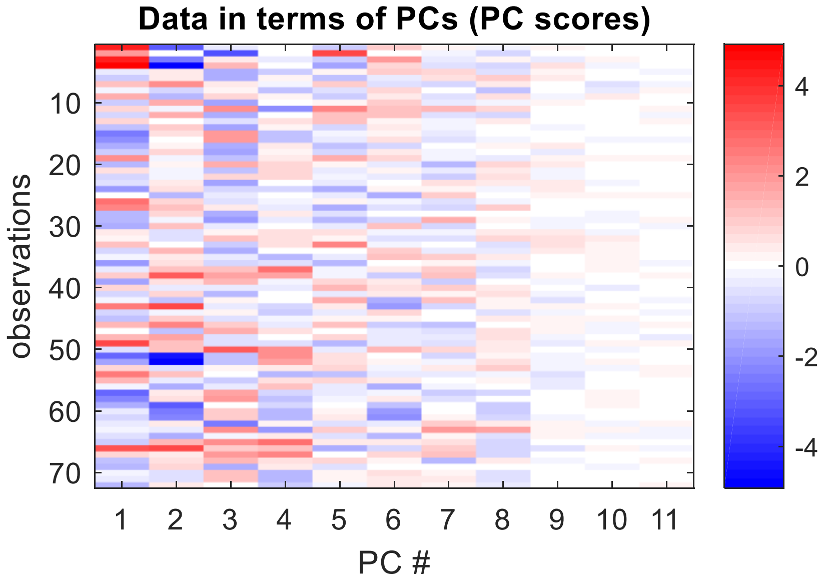
Express the data in terms of this new basis of principal components.

Tutorial

PCA

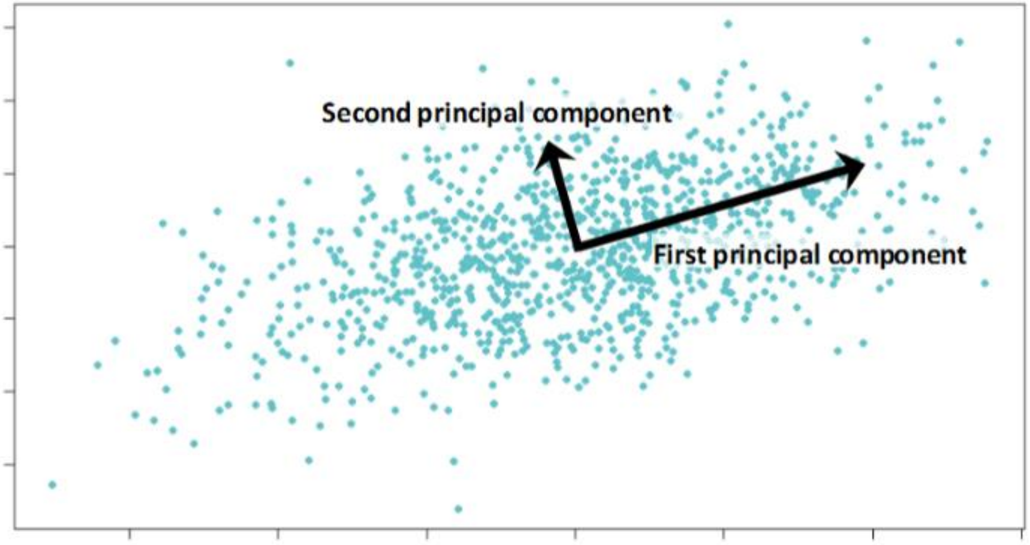
- Terminology:
  - **Loadings/coefficients:** how much each variable contributes to a PC
  - **Scores** (sometimes also called **PC loadings**!): how much each PC contributes to an observation





PCA properties

- PCs are **linear combinations** of the original variables (note: coefficients and scores can be negative!)
- PCs are **orthogonal**
- Number of PCs = the number of original variables, **p** (if n>p)
- First component captures direction of **maximal variability** in the original p dimensions (if data is centered first)
- Subsequent directions find directions along which the remaining variance is maximized, while still orthogonal to other components



A few reminders:

- Need to **center** the data (subtract mean value of each variable) in order for the first component to capture the direction of maximum variation
- **Scale matters:** if different variables have different units/scales, will need to z-score values of each variable first (check statistical function for default behaviour)
- **If n<p:** If we have fewer observations than variables (n < p), we can find n-1 PCs, rather than p PCs
- Various variations of PCA exist in order to overcome various limitations (e.g., sparse PCA, functional PCA, kernel PCA)

For dealing with n < p:
http://www.tandfonline.com/doi/abs/10.1198/jasa.2009.0121

Computing PCs without built-in pca functions

- **Preprocessing**: center or z-score data
- Compute **covariance matrix** or **correlation matrix**
  - Covariance matrix of z-scored data is equivalent to correlation matrix of the data before standardization
- Find **eigenvectors** of the matrix (→ PCs)
- **Project** preprocessed data into PC space
- Percentage of **variance explained** by each PC = eigenvalue/trace(covariance matrix)

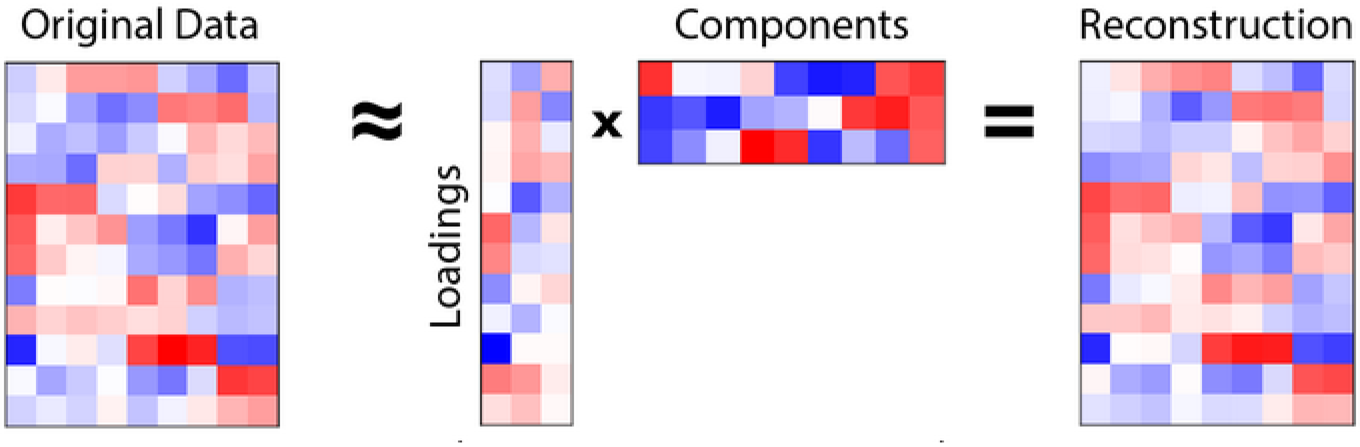
Computing PCs without built-in pca functions

- **Preprocessing**: center or z-score data
- Compute **covariance matrix** or **correlation matrix**
  - Covariance matrix of z-scored data is equivalent to correlation matrix of the data before standardization
- Find **eigenvectors** of the matrix (→ PCs)
- **Project** preprocessed data into PC space
- Percentage of **variance explained** by each PC = eigenvalue/trace(covariance matrix)

Tutorial

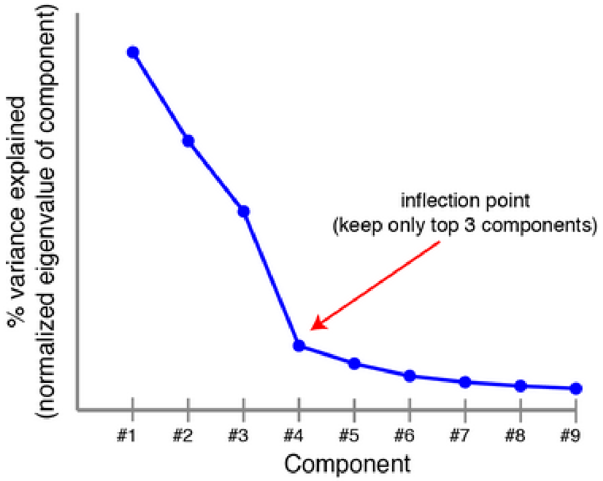
What do we do after PCA?

- Visualize data
- Determine how complex our data is (what is its dimensionality?)
- Reconstruct data with fewer dimensions, while (hopefully) keeping important information
- Project new observations into PC space

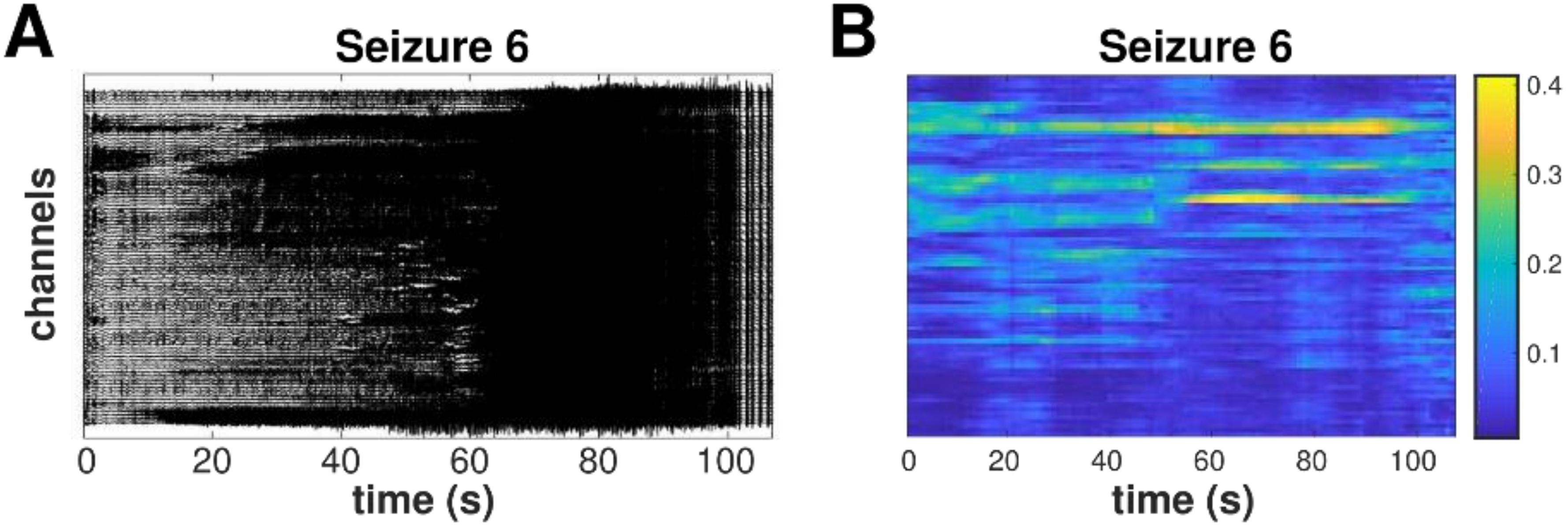


How many PCs should we use?

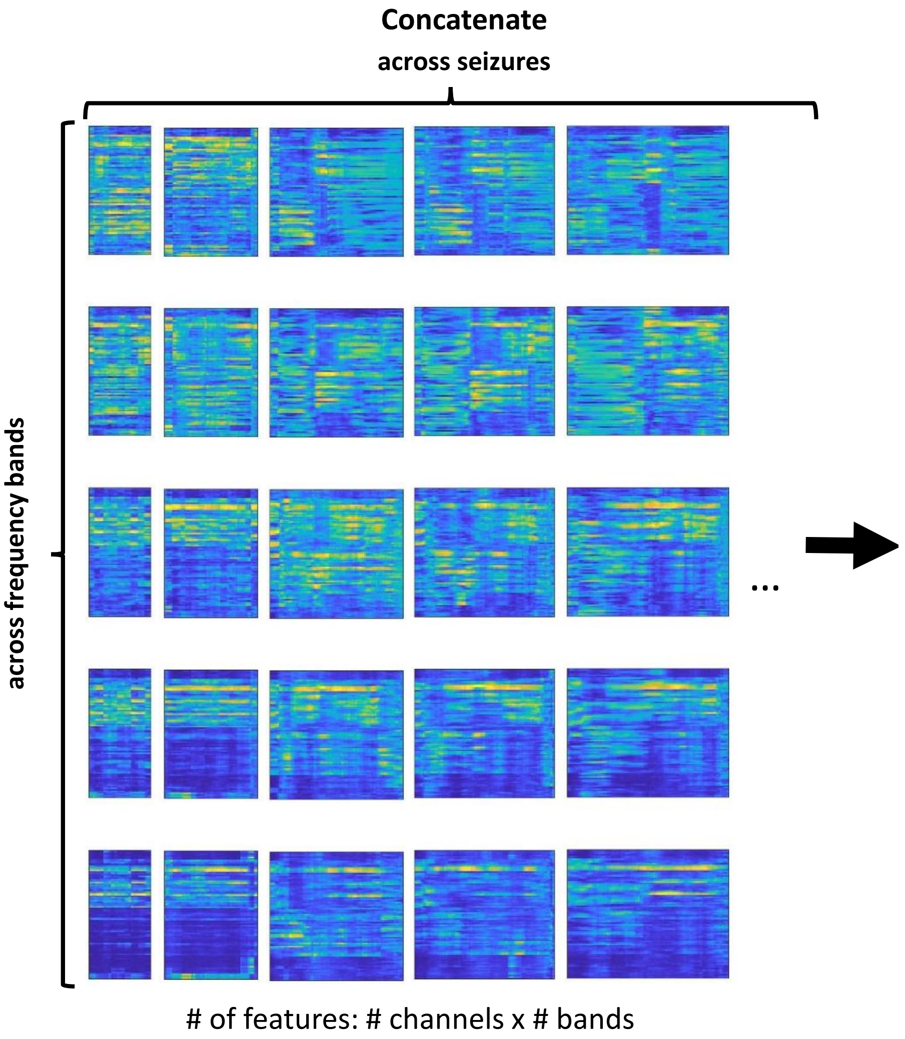
- Graphical limitations (easiest to visualize in two or three dimensions)
- Explain X% of total variance
- Remove PCs that capture noise
  - Determine based on scree plot
  - Estimate amount of noise in data (http://ieeexplore.ieee.org/document/6846297/)

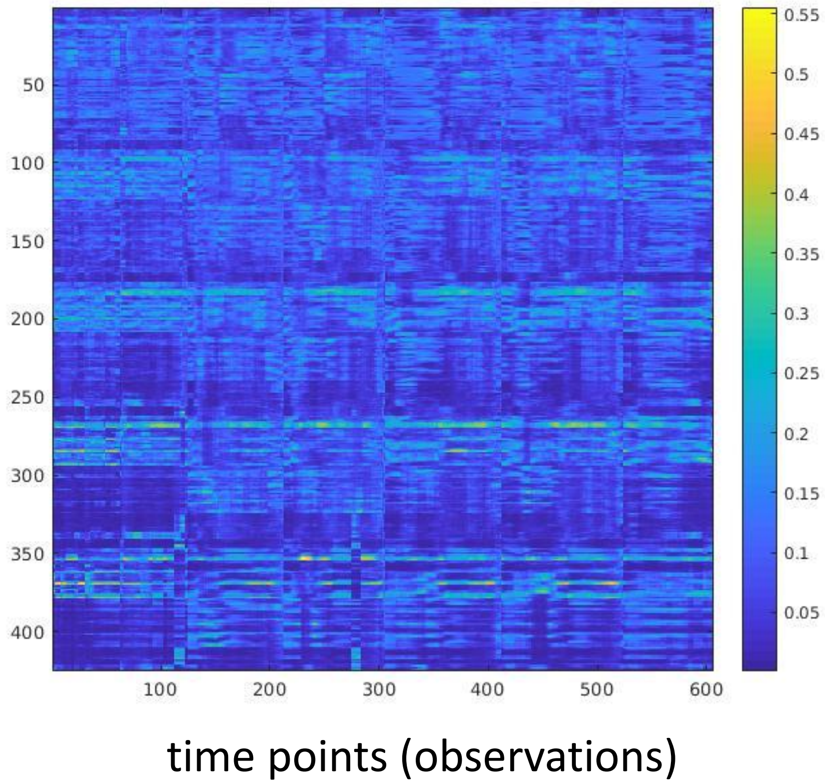


Neuroscience example: Visualizing seizure dynamics

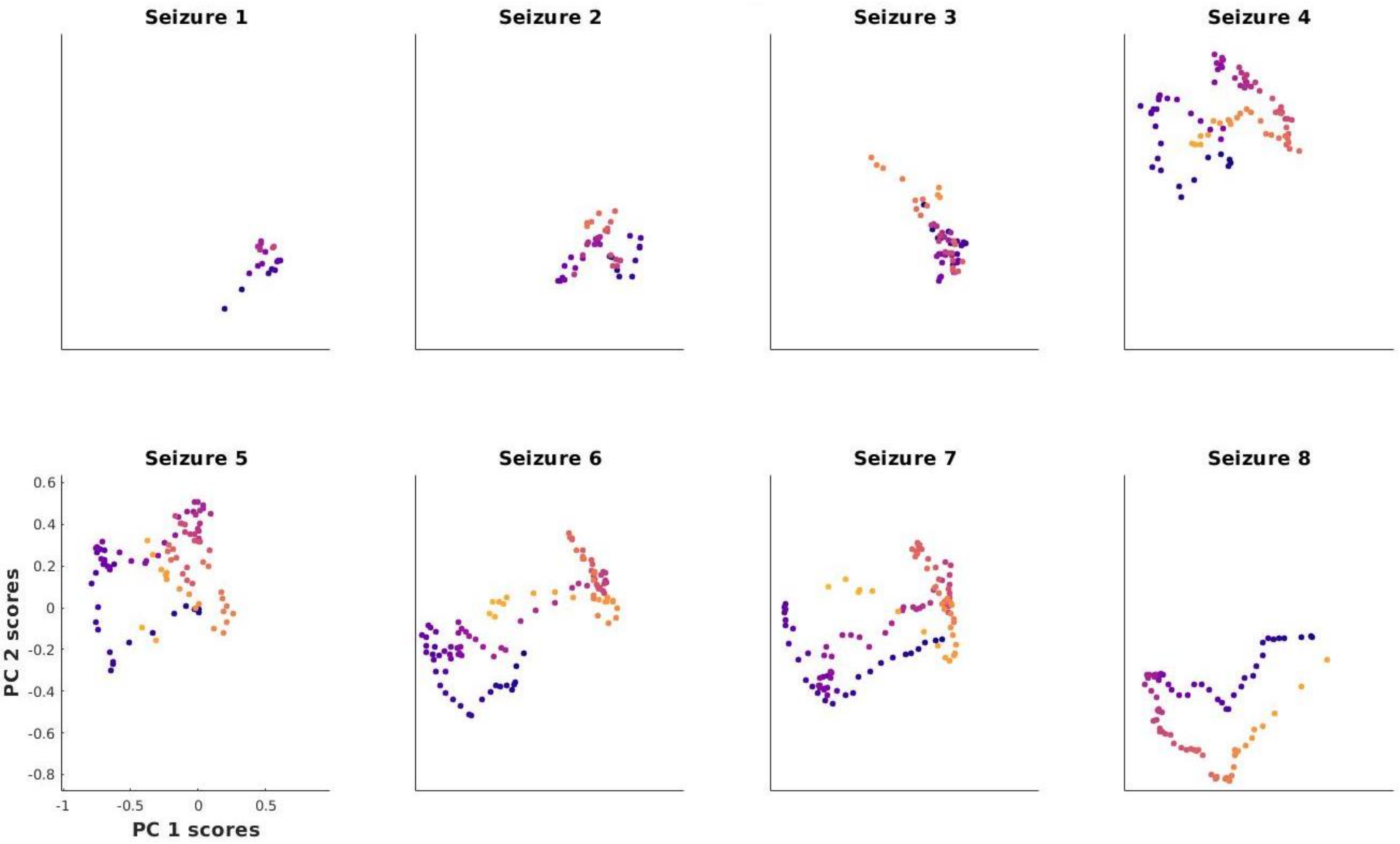


Neuroscience example: Visualizing seizure dynamics

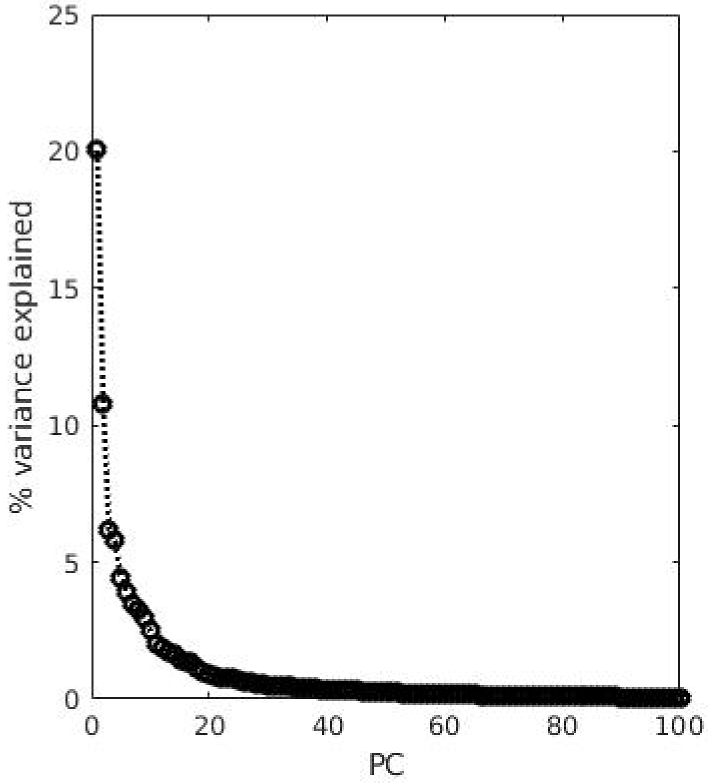


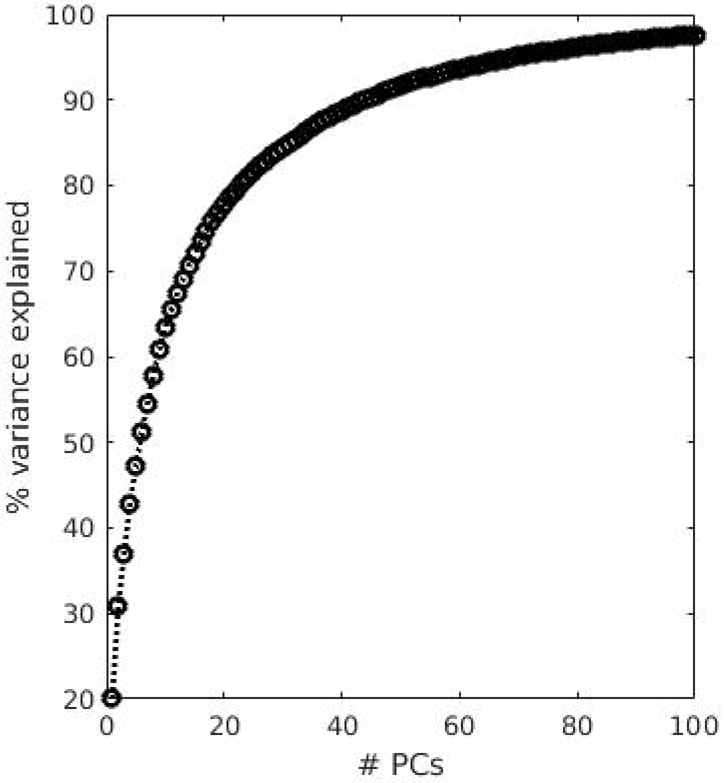


Neuroscience example: Visualizing seizure dynamics



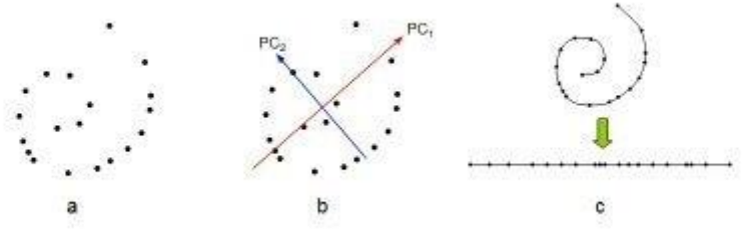
Neuroscience example: Visualizing seizure dynamics

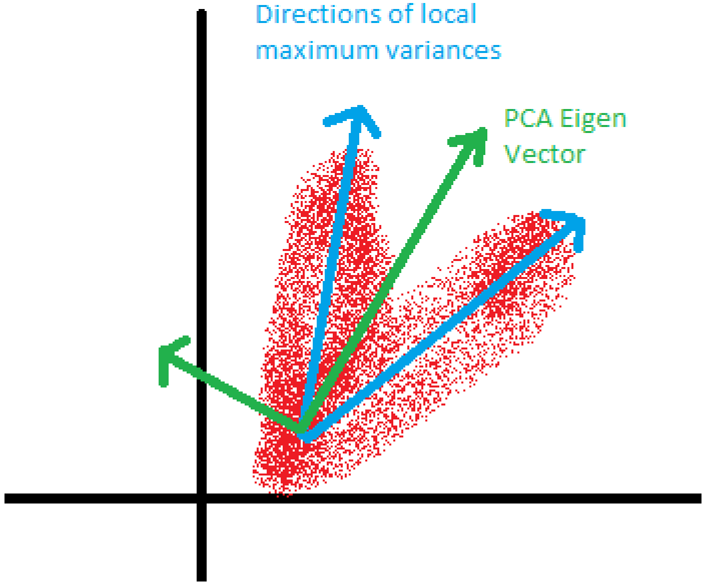




When PCA doesn’t work well…

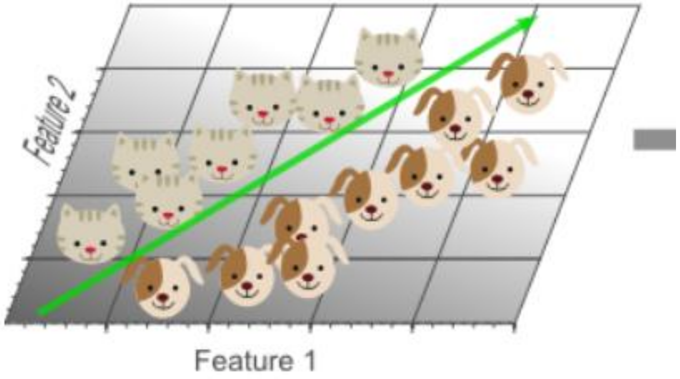
- Data is not linearly correlated
- Interesting directions may not be orthogonal
- Directions of maximum variance may not capture good features for distinguishing classes





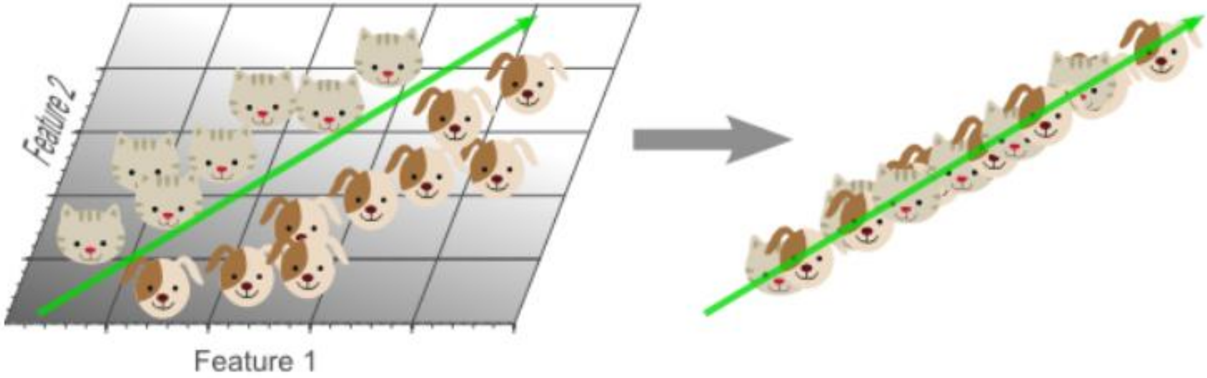
Reference: https://www.quora.com/What-are-some-of-the-limitations-of-principal-component-analysis

Caution: PCA (and other similar methods) may not find useful features for classification



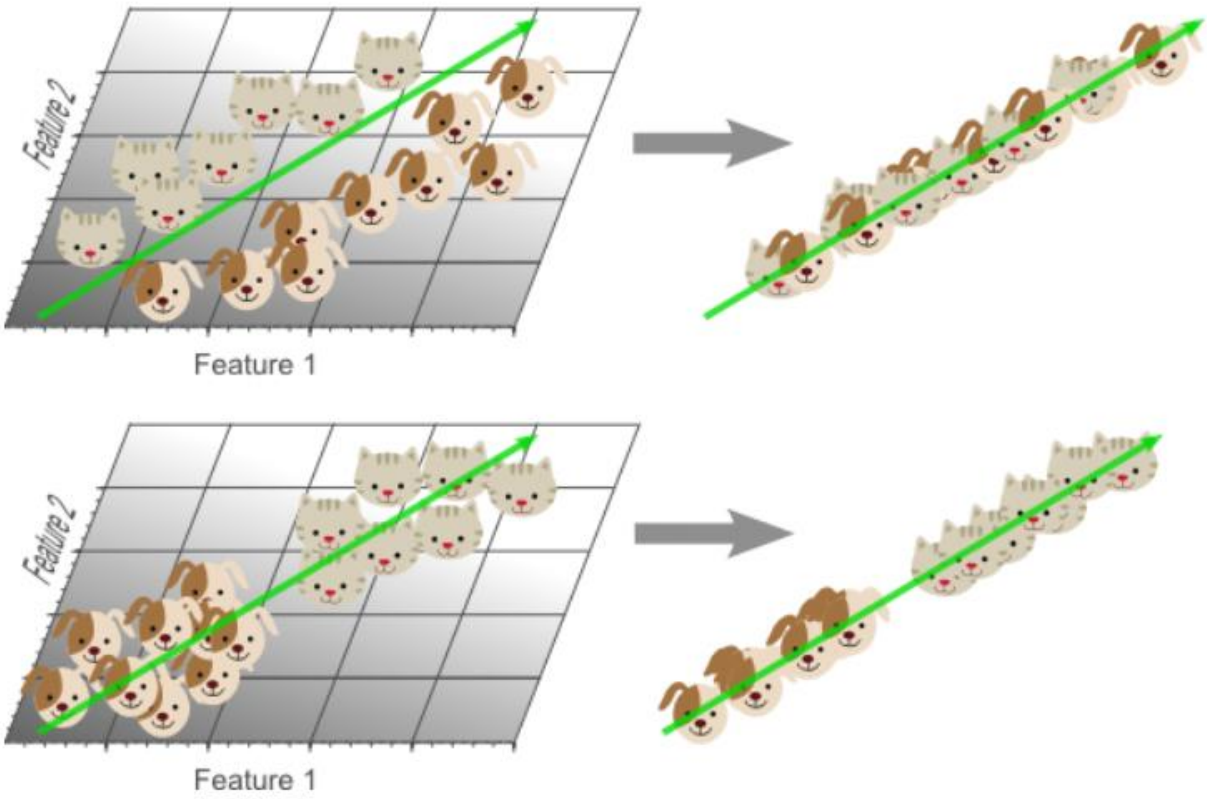
Reference: Kaggle forum: What is the proper way to use PCA? https://www.kaggle.com/general/21449#post122670 (also **How to not be dumb at applying Principal Component Analysis (PCA)?**: https://medium.com/data-design/how-to-not-be-dumb-at-applying-principal-component-analysis-pca-6c14de5b3c9d)

Caution: PCA (and other similar methods) may not find useful features for classification



Reference: Kaggle forum: What is the proper way to use PCA? https://www.kaggle.com/general/21449#post122670 (also **How to not be dumb at applying Principal Component Analysis (PCA)?**: https://medium.com/data-design/how-to-not-be-dumb-at-applying-principal-component-analysis-pca-6c14de5b3c9d)

Caution: PCA (and other similar methods) may not find useful features for classification



Reference: Kaggle forum: What is the proper way to use PCA? https://www.kaggle.com/general/21449#post122670 (also **How to not be dumb at applying Principal Component Analysis (PCA)?**: https://medium.com/data-design/how-to-not-be-dumb-at-applying-principal-component-analysis-pca-6c14de5b3c9d)

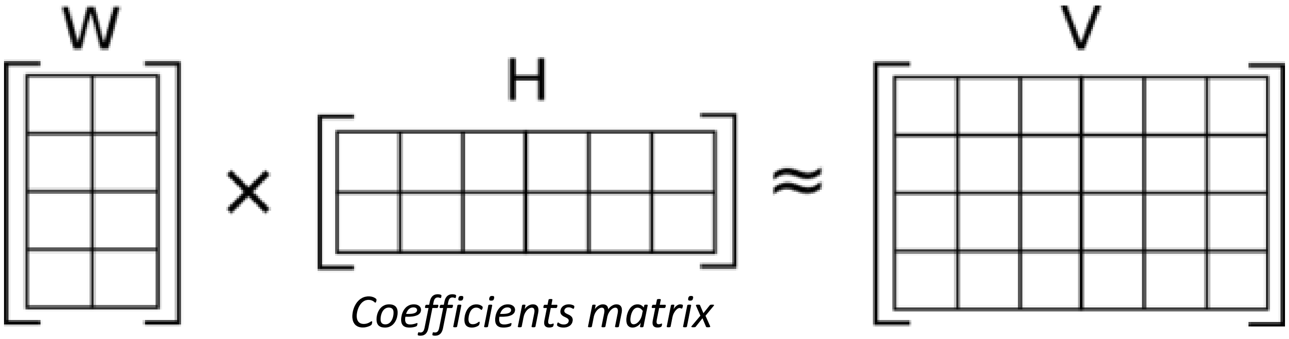
Non-negative matrix factorization

Non-negative matrix factorization (NMF)

- Decompose your matrix into two matrices, both of which can only contain non-negative entries

Basis matrix/features matrix
(each column = feature/basis vector)
2 features

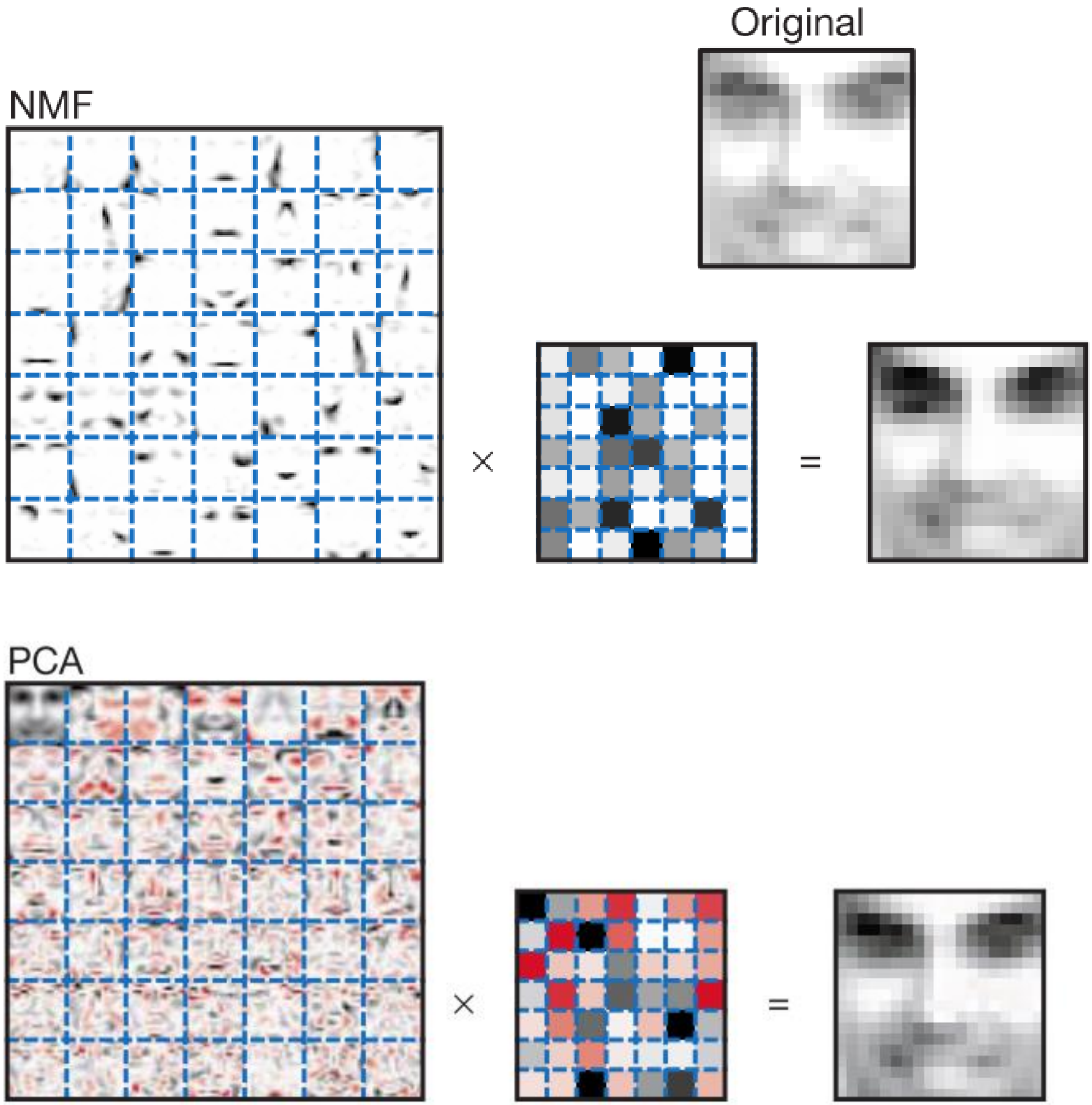
Original data
(each column = observation)
5 observations, 4 features



Coefficients matrix
(each column = observation)
5 observations

- **Reference:** Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401:788–791 Available at: http://www.ncbi.nlm.nih.gov/pubmed/10548103.
- **Matlab toolbox:** Li Y, Ngom A (2013) The non-negative matrix factorization toolbox for biological data mining. Source Code Biol Med 8:10 Available at: http://scfbm.biomedcentral.com/articles/10.1186/1751-0473-8-10.
- https://en.wikipedia.org/wiki/Non-negative_matrix_factorization

NMF



Reference: Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401:788–791 Available at: http://www.ncbi.nlm.nih.gov/pubmed/10548103.

NMF compared to PCA:

- No analytical solution (optimization problem)
  - Can find multiple good solutions; adding additional constraints (e.g., orthogonality, sparsity) can limit solutions
- Easier to interpret than PCA because components are non-negative and additive only; however, components are not ranked
- Choose the number of components at the beginning
- Only works for non-negative data (unless you use semi-NMF)
- Usually factor all of the data at the same time

Recurring Functional Interactions Predict Network Architecture of Interictal and Ictal States in Neocortical Epilepsy

Ankit N. Khambhati,[1,2] Danielle S. Bassett,[1,2,3] Brian S. Oommen,[2,4] Stephanie H. Chen,[2,4] Timothy H. Lucas,[2,5] Kathryn A. Davis,[2,4] and Brian Litt[1,2,4]

DOI:http://dx.doi.org/10.1523/ENEURO.0091-16.2017

[1]Department of Bioengineering, University of Pennsylvania, Philadelphia, PA 19104, [2]Penn Center for Neuroengineering and Therapeutics, University of Pennsylvania, Philadelphia, PA 19104, [3]Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, [4]Department of Neurology, Hospital of the University of Pennsylvania, Philadelphia, PA 19104, and [5]Department of Neurosurgery, Hospital of the University of Pennsylvania, Philadelphia, PA 19104

Abstract

Human epilepsy patients suffer from spontaneous seizures, which originate in brain regions that also subserve normal function. Prior studies demonstrate focal, neocortical epilepsy is associated with dysfunction, several hours before seizures. How does the epileptic network perpetuate dysfunction during baseline periods? To address this question, we developed an unsupervised machine learning technique to disentangle patterns of functional interactions between brain regions, or subgraphs, from dynamic functional networks constructed from approximately 100 h of intracranial recordings in each of 22 neocortical epilepsy patients. Using this approach, we found: (1) subgraphs from ictal (seizure) and interictal (baseline) epochs are topologically similar, (2) interictal subgraph topology and dynamics can predict brain regions that generate seizures, and (3) subgraphs undergo slower and more coordinated fluctuations during ictal epochs compared to interictal epochs. Our observations suggest that seizures mark a critical shift away from interictal states that is driven by changes in the dynamical expression of strongly interacting components of the epileptic network.

Key words: dynamic network neuroscience; epileptic network; non-negative matrix factorization; functional subgraphs; prediction; interictal

Other methods to be aware of…

Other methods

- **Singular value decomposition:** decomposes a matrix into three matrices

$$X = U\Sigma V$$

$$= \begin{pmatrix} | & | & & | \\ \vec{u}^{(1)} & \vec{u}^{(2)} & \cdots & \vec{u}^{(N)} \\ | & | & & | \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 & \cdots & 0 \\ 0 & 0 & \cdots & \lambda_N & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} - & \vec{v}^{(1)} & - \\ - & \vec{v}^{(2)} & - \\ & \vdots & \\ - & \vec{v}^{(N)} & - \\ & other & \end{pmatrix}$$

$$= \begin{pmatrix} | & | & & | \\ \vec{u}^{(1)} & \vec{u}^{(2)} & \cdots & \vec{u}^{(N)} \\ | & | & & | \end{pmatrix} \begin{pmatrix} - & \lambda_1 \vec{v}^{(1)} & - \\ - & \lambda_2 \vec{v}^{(2)} & - \\ & \vdots & \\ - & \lambda_N \vec{v}^{(N)} & - \end{pmatrix}$$

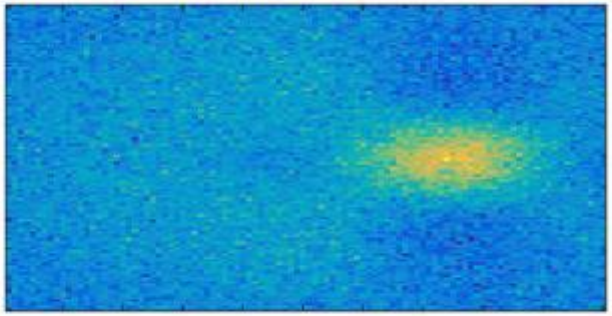
$$= \lambda_1 \begin{pmatrix} | \\ \vec{u}^{(1)} \\ | \end{pmatrix} \begin{pmatrix} - & \vec{v}^{(1)} & - \end{pmatrix} + \lambda_2 \begin{pmatrix} | \\ \vec{u}^{(2)} \\ | \end{pmatrix} \begin{pmatrix} - & \vec{v}^{(2)} & - \end{pmatrix} + \cdots + \lambda_N \begin{pmatrix} | \\ \vec{u}^{(N)} \\ | \end{pmatrix} \begin{pmatrix} - & \vec{v}^{(N)} & - \end{pmatrix}$$

SVD: https://web.stanford.edu/class/nbio228-01/ (Linear Algebra Lecture 2)

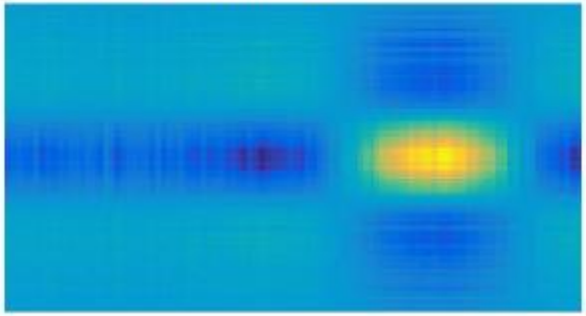
Other methods

Singular values are ordered by importance, and you can use the first singular values and the associated vectors to create a less noisy approximation of your original matrix:

Noisy receptive field:



Low-rank approximation:

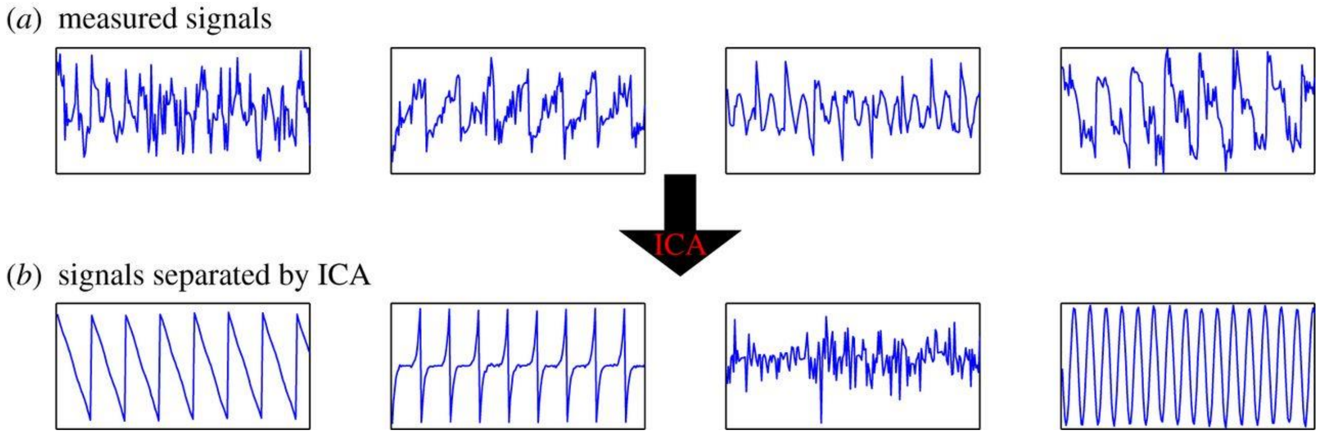


SVD is equivalent to PCA if the values for each variable are centered before the decomposition (this is how MATLAB performs PCA!)

SVD: https://web.stanford.edu/class/nbio228-01/ (Linear Algebra Lecture 2)
SVD vs PCA: https://stats.stackexchange.com/questions/134282/relationship-between-svd-and-pca-how-to-use-svd-to-perform-pca

Other methods

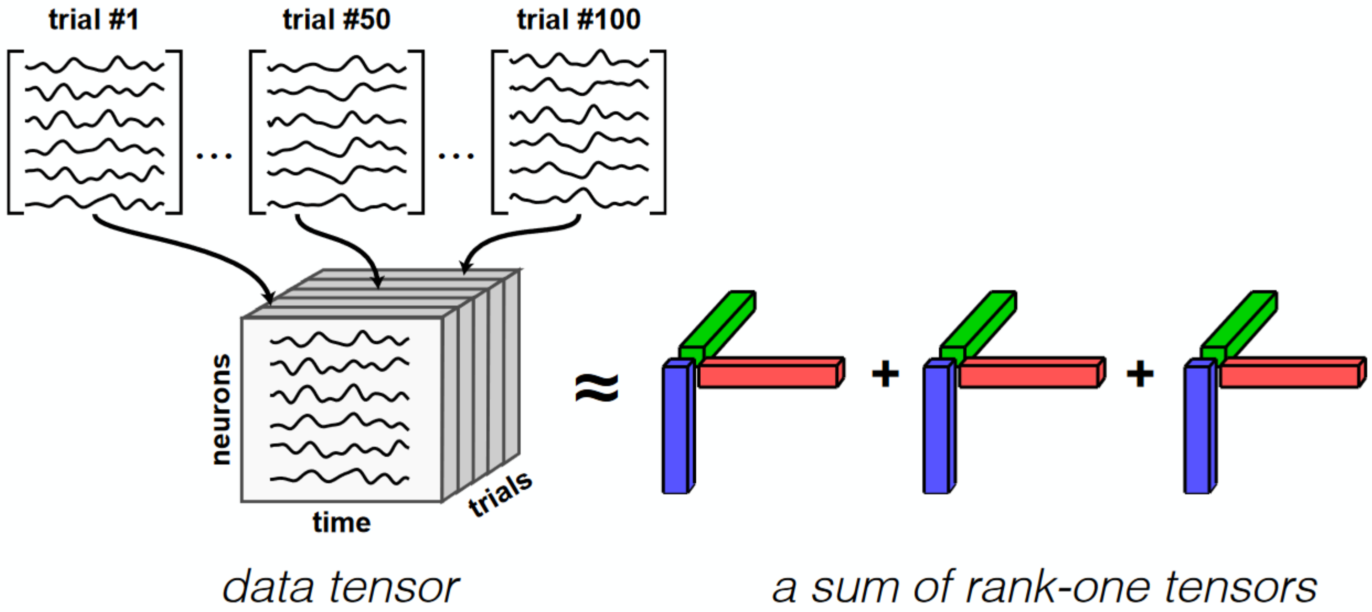
- **Linear discriminant analysis:** finds linear combinations of variables that best separates two+ classes
- **Independent components analysis:** find statistically independent components (e.g., “cocktail party problem”)



LDA: https://en.wikipedia.org/wiki/Linear_discriminant_analysis
ICA: https://en.wikipedia.org/wiki/Independent_component_analysis;
http://rsta.royalsocietypublishing.org/content/371/1984/20110534

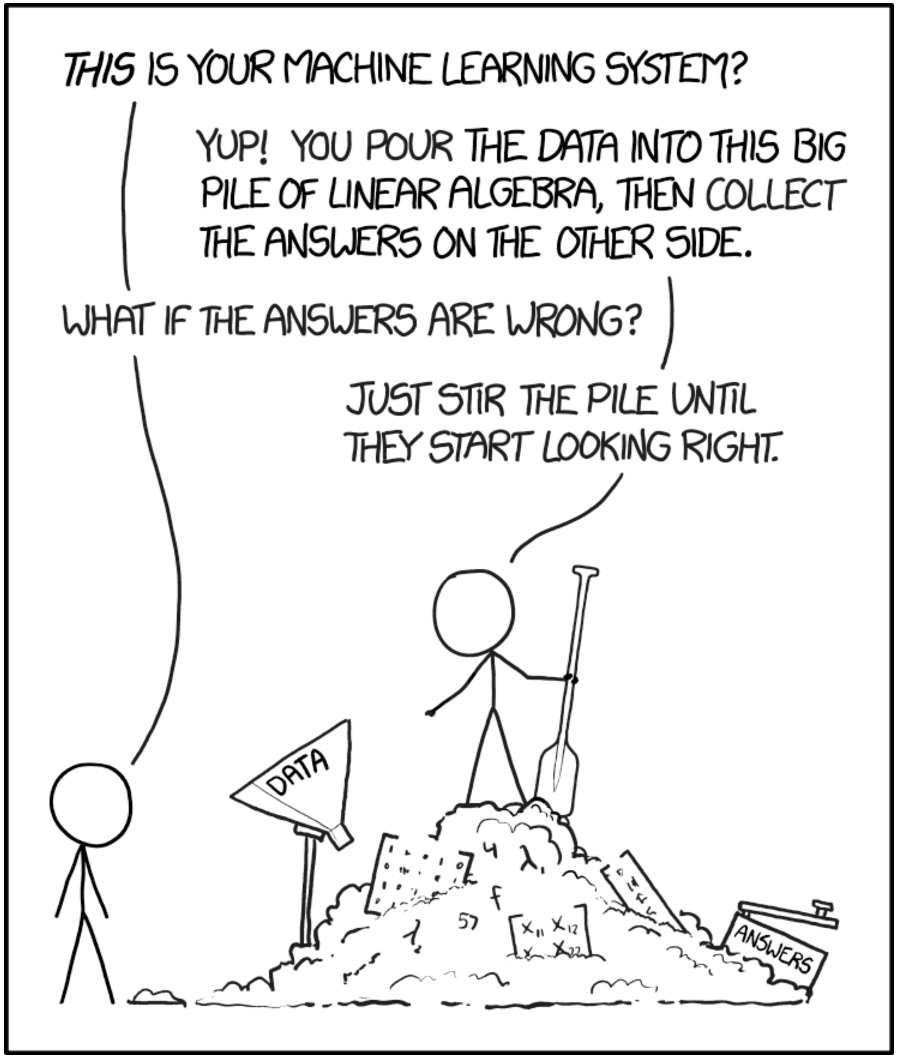
Other methods

- **Tensor decomposition:** generalizes SVD to tensors (= multidimensional arrays, could have >2 dimensions)



Tensor decomposition: https://en.wikipedia.org/wiki/Tensor_(intrinsic_definition); http://www.sandia.gov/~tgkolda/pubs/pubfiles/TensorReview.pdf; application: http://alexhwilliams.info/pdf/cosyne17.pdf

Questions?



THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Supplemental slides

Matrix times Matrix:
by outer products

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

$$\overleftrightarrow{C} = \begin{pmatrix} \\ A^{\mathrm{c1}} \\ \\ \end{pmatrix} \begin{pmatrix} & B^{\mathrm{r1}} & \end{pmatrix} + \begin{pmatrix} \\ A^{\mathrm{c2}} \\ \\ \end{pmatrix} \begin{pmatrix} & B^{\mathrm{r2}} & \end{pmatrix} + \cdots + \begin{pmatrix} \\ A^{\mathrm{cP}} \\ \\ \end{pmatrix} \begin{pmatrix} & B^{\mathrm{rP}} & \end{pmatrix}$$

- **C** is a sum of outer products of the columns of **A** with the rows of **B**

How to find eigenvalues and eigenvectors

Real life and easiest way: Using MATLAB's eig command

[V,D] = eig(W) will return a matrix V, where each column is an eigenvector, and a diagonal matrix D with the corresponding eigenvalues along the diagonal

MATLAB output:

$$W = \begin{bmatrix} 0 & 3 \\ 2 & 1 \end{bmatrix}$$

```
[V,D] = eig(W)

V =
  -0.8321   -0.7071
   0.5547   -0.7071

D =
   -2    0
    0    3
```

Eigenvectors are defined only up to a scale factor, and so they are typically scaled such that the norm of the vector is 1

Eigenvalue is 3, like we found before!

The set of eigenvalues is called the *spectrum* of a matrix