The book club is reading **Data Driven Science & Engineering: Machine Learning, Dynamical Systems, and Control** by Steven L. Brunton and J. Nathan Kutz

Overview

While you can get the book on Amazon and other places, the pages for this will be using the online book. PDF

The book has a lot of extras, its main page is https://databookuw.com/. There are problem sets there. The code:

- https://github.com/dynamicslab/databook_python
- https://github.com/dynamicslab/databook_matlab

The numpy version sometimes requires some additional configuration. There is a good chance I (Elecia) have a running colab. Mathworks also provides 20 hours of Matlab on file for free per month. And some have used Octave for the matlab examples.

The goal is 15-25 pages/week. Note that there are different sections so you can drop in later.

Basically, read half a chapter every week. We'll discuss as we go on if this is a manageable pace. Where applicable, the associated videos will be noted from the author's YouTube channel.

Note that we have Linda Patton joining us and as she's a math prof and she's getting some tutoring bucks for this, ask questions!

Discussions will be live as well as on the slack. Tentative time is Fridays 11am. (Calendar invite)

Overview Links

(Stuff I had to look up more than once)

- The Matrix Cookbook: 60 pages of matrix cheatsheat. Dense, for reminding not learning.

Week 1: Ch 1 Singular Value Decomposition Part 1

Read through 1.5 PCA.
The first set of videos: Singular Value Decomposition (SVD): Overview
Meet June 20 (link)

This is a lot of math. Don't be afraid! Write down your questions as comments here or in slack.

Matrix Math help:

- High level matrix math (linear algebra) cheat sheet. Covers basics, does not get into the fancy terms
- Here is a book with useful sample chapters that do cover the fancy terms (Matrix Analysis by Roger Horn and Charles Johnson)
- Free recommended book! Linear algebra and its applications by Gilbert Strang (Homepage) and MIT online course about Linear

Housekeeping

- Meet Linda Patton
- Agenda definition: housekeeping, open q, disc q, vid or run code?
- Get everyone invited to the calendar link
- Anyone missing links to book, book website, vids, slack #book-club?
- 4th of July?

Some discussion questions:

- Given the table of contents in the book, have you read books with these topics that you like? Machine learning? Controls?
- CS vs Physics and Engineering: different techniques- who is in which camp?
- Folks who watched the vids: which one or two was most useful? (There is a good page on the book's website to see which videos relate to which chapters.)
- Folks who ran the code: anything cool?
- Anything that made you look at something you know differently?
    - El: The idea that DFTs are SVDs. The image problem (as I think in time series data).

Open questions:

- Tom: complex numbers. Seems like maybe some of the X^2 needs to be X*complex_conjugate(X)?

Hermitian aka self adjoint: symmetric if real
Here is a real hermitian matrix

```
RealH = [ [ 1 2 ]
            2  3 ] ]
```

Note that the top right and bottom left are symmetric around the diagonal.

For a complex Hermitian matrix:
```
ComplexH = [ [ 1      2+2i]
             [ 2-2i   3    ] ]
```

The top right and bottom left are complex conjugates

- **Rank** is the number of linearly independent columns. Or rows. Here is an example matrix
  ```
  mtx = [ [ 1 1 1 1]
          [ 1 1 0 0]
          [ 1 0 1 0]
          [ 1 1 1 1] ]
  ```
  While it is not obvious that this is rank 3 in the columns because they are all different, looking at the rows, it is more obvious because the top and bottom rows are the same.

  MAGIC: The number of independent columns is the same as the number of rows
  This works for any matrix, not just square ones.

  Maximum rank is the minimum of the number of row and the number of columns

  Consider **rank** in the context of data is **dimensionality**: how many basis vectors do I need to construct any column in the matrix? This lends itself nicely to thinking about PCA.

- **Unitary**: orthonormal and length one, AA* = I (but not A*A = I), right?
  - Identity matrix is the best example of a unitary matix
  - Like the XYZ coordinate matrix
  - If AB = I, the BA = I for a square matrix A (mxm)
  - Unitary means it is always square
  - AA* = I AND A*A=I as long as it is square!
  - SVD **A** = **U** ∑ **V*** then approximate **A** with a lower rank matrix (throw away the boring, low value data). This may make ∑ not a square!
  - **A** and ∑ are the same size (before rank reduction)
- **Unitary before rank reduction**
  - **A** is (m by n). **U** is (m by m). ∑ is (m by n). **V** is (n by. n)
  - A=U∑ V^T
- **After rank reduction from n to r.**
  - If you only want r singular values (big ones),
  - Then A is approximately U (m by r), ∑ (r by r) V^T (r by n)

    - U and V end up being unitary… but if you reduce the rank, they won't be square and can't be unitary
    - In this case, UU* and U*U may not equal

A matrix's **rank** is the number of linearly independent columns it contains. Column rank and row rank are equivalent (because of some theorem)

How to choose the desired rank? This video answers the question best. But here are some thoughts:

- It depends on what you are optimizing for. Consider the dog example in the book. Lower rank made for much less data but also less clarity of the picture.
- However, with lots of data, there is often a point where the sigma values are below the noise floor (See how the rank 100 dog pic looked almost the same as the original?)
- Lower rank means less data.
    - Each additional rank after the previous one gives diminishing returns on your approximation of the original
- You can plot the sigma to see where the energy (goodness) of the rank drops off (Fig 1.4 in the book)

Best notes/quotes:

- Kevin on recalling linear algebra: Yeah, I'm having to call back penguins that left the iceberg a LONG time ago.

Takeaways

- SVD is a powerful alternative to linear regression. I did learn in school the least squares regression (w = (X.T @ X)^-1 @ X @ y) but I learned that SVD allowing you to take pseudo-inverses means you don't need to worry about X.T @ X being singular.
- You can reduce the amount of 'storage' of data using SVD - the dog photo example was really cool to see. Curious to see how this can be applied.
- Look at the Jupyter notebook for the Ovarian Cancer dataset. He covers the 'practical' math needed for the PCA there.
    - Perform the SVD on the observations matrix.
    - Take the dot product of the observation vectors (rows of the observation matrix) with the rows of VT. Each row of VT is a principal component.
- Unitary matrix (wiki) is cool but… The full U (which is square mxm) from the SVD satisfies U*U=UU*=I. The reduced rank U tilde is an mxr matrix. U*U will be the rxr identity matrix, but U U^* will be an mxm matrix that is not I because U's rows are no longer orthogonal to each other (the last m-r entries are removed).
- If the original matrix is real, so is U. Then U*=U^T. If original is complex, you need the conjugate transpose U*, aka the adjoint.

Week 2: Ch1 SVD Part 2

Re-read PCA section, watch vids on PCA, they are better than the text.
Finish Ch 1. (A little shorter than last week)
Meet June 27 (link)

From the book: **The Optimal Hard Threshold for Singular Values is 4/sqrt(3)** by Matan Gavish, David L. Donoho 2014: PDF here, Code tarball here

Some discussion questions:

- My mind was blown by the denoising in the toy problem, that lower r would be better. Did anyone else have any “wait, what?” moments?
- The code… their code is good and useful. So much code is not. Is it only useful because it is well explained?
- When to use SVD and when not to? When to use PCA instead of SVD?
- Did anyone try any of these techniques on their own data?
- “The SVD is only generically invariant to unitary transformations, meaning that the transformation preserves the inner product”: what does this mean and why does it matter?
- The random sampling made sense to me, maybe I don’t understand it?

What is a tensor?
Vector is one column
Matrix is an array (mxn: m columns, n rows)
Tensor is a cube (or higher order): (m x n x p)

> Animation: pixel x and y, then time: as a data set this is a box
> Addition makes sense
> Different kinds of multiplication (needs to be have definition)

To get colab working for CH01_SEC09_Tensor, python 3.8.5 & matplotlib 3.3 was required.
Scott also found a code change

```
In the end, a code change was also required -
B1, B2 = parafac(A,2)
A1 = B2[0]
A2 = B2[1]
A3 = B2[2]
```

Video for Tensor Decomposition was the coauthor Kutz: Applied Linear Algebra: Tensor Decompositions. It is a bit longer than some of the other videos (at 40 min).

The green blob from CH01_SEC09_Tensor code is better shown around minute 20 of the meshgrid example in the **second** Kutz video Applied Linear Algebra: Implementing Tensor Decompositions

-> Linda will recommend a great tensor description

PCA (again)
Linda posted a PCA matlab examples with students taking quizzes. There were 10 students (columns) and 5 questions (rows). All students got question 5 right so it wasn't a good differentiator of student's understanding. Student 1 got all the questions correct so that keener was breaking the curve. The curve is represented in the sigma of the SVD which was similar to adding up the student scores over all of the questions (except the too-easy question 5).

PCA was better at determining which questions were most useful. Questions that everyone got right or that almost everyone got wrong weren't useful.

PCA as a way to optimize data needed.

Week 3: Ch2 Fourier Pt 1

Through section 2.3 Transforming PDEs
Meet July 4 (link)

Most interesting fact from this section

Gauss discovered an algorithm for computing the DFT in 1805, before Fourier even published his work on functions as harmonic series. The Wikipedia page has a nice summary, but this paper provides more interesting details.

Discussion questions:

- How much of this have you seen? Could you learn these topics from this book?
- Kappa vs omega: spatial (wavenumber) vs temporal waves
- FFT as a basis! As a coordinate system!

Notes:

- Burgers' Equation has a nice wiki page with gif-graphs
- Fourier properties in wiki
- DFTs only came out in 1965! Cooley-Tukey were working on finding nuclear explosions.
    - The Most Important Algorithm Of All Time

Next time: Windowing: how do we know what the valid time window is after we return from frequency space?

Week 4: Ch2 Fourier (finish)

Finish Ch2
Meet July 11 (link)

Most interesting fact from this section:
The time-frequency uncertainty principle is the same uncertainty behind the Heisenburg uncertainty principle.

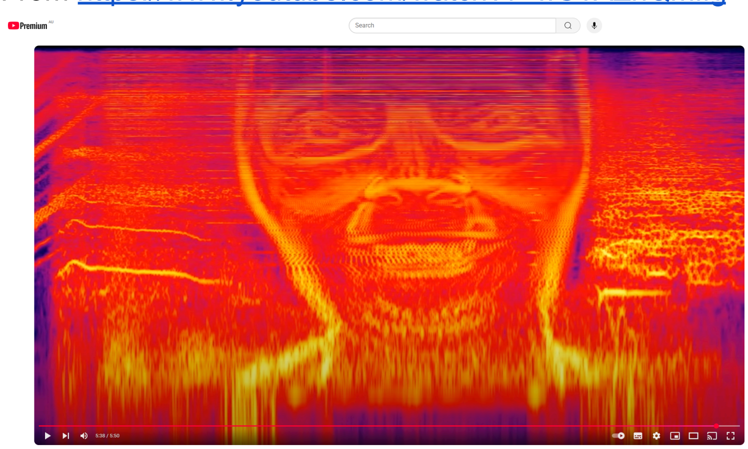
Discussion questions:
- Has anyone used wavelets in the wild? ECGs are the traditional example (Wavelet Analysis of ECG)
- Any good Fourier or Wavelet resources online?
- SVD (Specrotgram): Eigenchords, what else can this apply to?

Notes:
There is a homework assignment for this section: Analyze Handel's Messiah (from the problem set page).

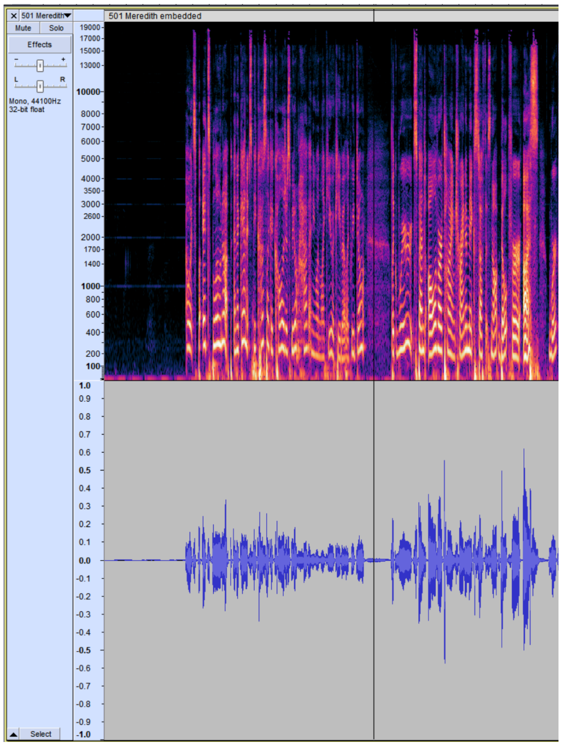
Whales and spectrogram https://www.youtube.com/watch?v=5tRMqbPH_pk
https://www.mbari.org/project/soundscape-listening-room/

A Wavelet Tour of Signal Processing: The Sparse Way by Mallat, Stephane
(book piracy link) https://libgen.li/edition.php?id=138736425

From https://www.youtube.com/watch?v=wSYAZnQmffg -



Audacity has spectrogram, click on the track name



Week 5: Ch3: Sparsity and Compressed Sensing

Ch 3 through 3.5
Meet July 18 (link)

Most interesting fact from this section:

- The vastness of pixel space compared to the amount of space taken by actually useful images (Image 3.3). Most of the possible images are simply noise.
- Nyquist means you **must** sample at 2x the highest frequency of your data.
  - Well, actually… if you sample randomly (given a known sample rate)....
- Elecia likes the word parsimonious and the idea models should be interpretable

Discussion questions:

- This seems really broken. I can't figure out how to use this on an image! (I will try it, possibly with the Compressed Sensing repo using MNIST measurements data. -El)
- Parts of this are computationally expensive, so how do we use it?
  - Determining the sparse vector of s is hard, once you have that, your smaller number of measurement samples (y) can lead to a reconstruction of the original signal (x)
  - Does this mean the sparse vector (s) is not general? That it limits the rebuilding of the original signal to certain cases?
  - "Compressed Sensing: When It Works" (video start time is when he mentions number of measurements likely required)

- This all seems to be leading us to machine learning.

Notes

- **Library of Babel** by Jorge Borges
- Compressed sensing - Wikipedia
- Imaging objects out of sight using a single photodetector (From Mark in slack: video of a camera using compressed sensing)
- Zero padding gives you more frequency resolution without getting more samples, discussed in detail in https://www.dspguide.com/CH14.PDF, demonstrated with a nice slider: https://jackschaedler.github.io/circles-sines-signals/zeropadding.html
    - Zero padding seems like an alternative to compressed sensing

Week 6: Finish Ch3

Ch 3 through 3.5
Meet July 25 (link)

Most interesting facts from this section:

- Use of the word **sensor** was odd in this section, in the images it was a pixel. In the time series data it was a measurement. A sensor was a measurement.
- L1 norms and taxi cab geometry: Tom recommended Taxicab Geometry: An Adventure in Non-Euclidean Geometry (Dover Books on Mathematics): Krause, Eugene F.
- LASSO: don't let data wonks push you around, as for sparse models (and interpretable)

Discussion questions:

- Better ideas about 2d compressed sensing?
- Sensor location and the C matrix, let's talk examples
- What are the best ways you have to describe eigenvectors? (Eigenstamps) How does that apply to compressed sensing?

Notes:

- Ben at Applied Science had a video posted this week: X-ray backscatter with compressed sensing algorithm. Even better, his references for the video included a link to the original inspiration from PyRunner.com Blog which had good code in their Compressed Sensing in Python post (Wayback) (better than in the course notebook). Ben also put up his code in github.
- Given these algorithms, rotation invariance seems like a problem (Hough transform - Wikipedia). Translation and scaling as well.
    - Probably in the ML chapter

- Hristo: Here's a Colab notebook if you're interested in IMU-gesture letters. The data itself is here (a description is in the notebook).

Week 7: Start Ch 4

Read through chapter 4.4
Meet Aug 1 (link)

Videos are from Kutz this time. While you can always find the relevant video for each section at databookuw.com, sometimes the playlist is tricky to locate. Scott found the playlist for Ch4.

Quote - “if you don’t cross-validate, you is dumb”

“I’m going to show you six different ways to solve Ax=b in Matlab” - Kutz… “Six different ways to solve Ax=b” seems like a good album name.

Most interesting facts from this section:

- R is the language formerly known as S. (Statistical Models in S)
- Optimization is the cornerstone of regression. Regression is the cornerstone of ML.

Discussion questions:

- Some repetition is good, was this good? What parts make more sense given this section?
- Are Figs 4.8 and 4.11 different? How do they represent over and under determined? Do you have a good grasp on the difference?
- Clearly Brunton and Kutz have different styles in writing and lecturing. Which do you like better? Why? Do you struggle against some times of teaching?
- Figure 4.13 was slightly infuriating to me… noise dependent models, have you seen any? It is one of my most worrisome things about ML: the lack of interpretability. Fig 4.15 was startling and made me want to see when the other methodologies are useful. We started dissing least squares for outliers but it is failing on small noise as well.

Notes:

- Datasaurus dozen: a collection of different small data sets that have the same summary statistics. You can see all the graphs here. From Humble Pi: A Comedy of Maths Errors.
    - What stats would the datasaurus dozen fall to? Maybe Chi-squared distribution.
    -
- I Saved a PNG Image To A Bird YouTube video was really neat, led to AudioMoth ultrasonic recorders as well as making a $40 bird identifier with an RPi and some software (BirdNET-Pi).
- Some of Tom’s curves and books

- CRC Handbook of Mathematical Curves & Surfaces: von Seggern, David H. Note there is a Mathematic one with generating functions
- Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables by Milton Abramowitz, Irene A. Stegun (also online, uncertain provenance)
- Practical Handbook of Curve Design and Generation (for art?) Google books has a nice sample
- Red book (Internet Archive PDF): Handbook of Filter Synthesis by Anatol I. Zverev

Week 8: Finish Ch 4: Regression and Model Selection

Read through the end of chapter 4, vids are in the playlist for Ch4.
Meet Aug 8 (link)

Most interesting facts from this section:

- Fancy Math Anonymous
- "William of Occam (c. 1287–1347), who was an English Franciscan friar, scholastic philosopher, and theologian. Occam proposed his law of parsimony (in latin lex parsimoniae), commonly known as Occam's razor, whereby he stated that among competing hypotheses, the one with the fewest assumptions should be selected, or when you have two competing theories that make exactly the same predictions, the simpler one is the more likely."
-

Discussion questions:

- What do you hope to get from this book? What are you getting?
- Anyone using other methods of explaining things? "In the context of data science, explain the pareto principle" Gemini.

Notes:

- Related to weight averaging in cross-validation, there's this idea of "model souping" proposed in this paper. The interesting part here is that the authors propose weight averaging between models from different training runs/different initializations. This is somewhat unusual, because typically we assume that the parameter space of large neural networks is so large that there's many local optima that we might end up with, and there's no reason to expect that averaging weights between different models would lead to a sensible outcome. And yet it sometimes works. - Hristo
- More relevant to next week: https://caffeineandlasers.com/blogs/Blog-Statsforfunscienceandart.html
- Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series)

- Hristo’s IMU: looking at IMUs in frequency domain
- Tom sent an Excel cookbook for filter design: https://github.com/loudifier/Biquad-Cookbook

Week 9: Classification! Ch 5: through Dendograms

Read through 5.5 Mixture models and the expectation-maximization algorithm
Meet Aug 15 (link)

Most interesting facts from this section:

- Iris species data set and telling them apart
- (5.5) Derivatives not being available
    - Automatic differentiation: read code or a graph (think Tensorflow) and automatically create analytic differentiation solution to aid in gradient descent (or as a feature)
- Dendrograms are neat!

Notes:

- If you're musically inclined, you might be interested in this collection of collab notebooks that analyze different aspects of music—tempo, genre and what have you. It uses some of the same techniques we've been looking at in the book! (Hristo)
    - Some of the data and libraries are out of date but searching for the errors let to working version (you may need to provide your own audio though)
- Nicely animated clustering algorithms: Clustering with Scikit with GIFs - dashee87.github.io
- Note that Scikit Learn (sk-learn) and Mathworks both have sites with code and examples about the topics covered in the book.
    - What Is Unsupervised Learning? - MATLAB & Simulink
    - 13. Choosing the right estimator — scikit-learn 1.7.1 documentation
- Tom wondered why the dendrogram focused on the distance from each point to each point when there is a different way:
    - Create a triangle mesh
    - Find the distance to the nearest triangle edge
    - Which should be a simpler calculation
    - From Professor Jonathan Shewchuk at Berkeley (and CMU)
    - Triangle: A Two-Dimensional Quality Mesh Generator and Delaunay Triangulator
    - Triangle: Demonstration
    - Voronoi objects are involved here but we didn’t follow the path
- I (Elecia) went from K-means to origami in two steps!
    - Lloyd’d algorithm for K-means clustering
    - Leads to Voronoi diagrams
    - Leads to any 3d shape from origami

- Kaggle. Contests and dataset. Get a dataset, try out some techniques. Look for datasets with usability 10. Or look at other notebooks. Lots of different data!
-
- Semantic clustering unsupervised learning: natural language processing and machine translation
  - King – Man + Woman = Queen: The Marvelous Mathematics of Computational Linguistics | MIT Technology Review
  - What Are Word Embeddings? | IBM
  - Clustering and Visualising Documents using Word Embeddings | Programming Historian
  - https://vpekar.github.io/semantic-clustering-of-words-with-majorclust-and-word2vec.html

Week 10: Finish Ch 5

Read through end of chapter 5
Meet Aug 22 (link)

Most interesting facts from this section:
- Decision trees are flow charts

Discussion questions:
- Figure 5.20: did it surprise you?
- What would you use for what? Look through Kaggle again?
- Glossing over Bayes like that, who was offended?

Notes:
- Kernel tricks confuse me (Elecia)
  - SVM Kernels : Data Science Concepts : ends up with “this is magical”
  - SVM Kernel Trick in 10 minutes : explain computational reasons which is nice (start around minute 4)
- DuckDB: Open source SQL
- Some books
  - Pattern Recognition and Machine Learning (Information Science and Statistics) by Chris Bishop
  - Pattern Classification: Duda, Richard O., Hart, Peter E., Stork, David G : 2001, algorithms have changed and some have become more popular
- Tom’s database uses strsimpy · PyPI to find the distance between strings (MetricLCS)

    - Description of distance metrics: GitHub - feature23/StringSimilarity.NET: A .NET port of java-string-similarity also see String metric - Wikipedia
        - Rob points out that some of these same algorithms can be used for genetics for insertions and deletions, alignment

Week 11: Finish Ch 6

Read through end of chapter 6
Meet Aug 29 (link)

Most interesting facts from this section:

- "Neural networks (NNs) were inspired by the Nobel prize winning work of Hubel and Wiesel on the primary visual cortex of cats." https://pmc.ncbi.nlm.nih.gov/articles/PMC1359523/

Discussion questions:

- Was this a terrible introduction to backprop?
- Activation functions! How do you choose? What network configuration do you choose? (No one has the definitive answer, the book says, "But as much as we would like to have a principled approach to building DCNNs, there remains a great deal of artistry and expert intuition for producing the highest performing networks."

Notes:

- The Neural Network Zoo - The Asimov Institute
- Lorenz system: Why the Lorenz 63? It is well-studied chaotic system, dealing with weather convection, extremely sensitive to initial conditions. It is a strange attractor: interesting because things tend to go to a specific pattern but the initial conditions mean you don't know when they get where.
- Activation function - Wikipedia How do you choose activation functions? Magic?
- 8.1. Toy datasets — scikit-learn 1.7.1 documentation: While Kaggle is interesting, sometimes. Wikipedia also has lists of data: List of datasets for machine-learning research - Wikipedia
- Deep Learning by Ian Goodfellow and Yoshua Bengio and Aaron Courville: how to design deep learning system; a lot of math

Week 11: Read through Ch 7.3 (SINDY)

Read through end of chapter 7.3
Meet Sept 5 (link)

Most interesting facts from this section:

- ScottH: The idea that you can apply what looks like the techniques from chapter 1 to CFD and get interesting results is blowing my little mind. Add in that you can extract the underlying equations of such a complex system using data techniques - I know a guy who spent a year solving Navier-Stokes by hand. Am boggled.
- DMD is super cool… but not easy to understand.
- Is DMD a reinvention of Lyapunov stability? Why isn't this referenced?

Discussion questions:

- Can we make a quiz?
- Why do we keep making estimators? How do we get from here to control? Is this about stability (linearization) or about control?
-

Notes:

- ScottH: https://github.com/python-control/python-control/issues/847 Jumped ahead to chapter 9 code example because I was curious - this will be relevant in a couple of weeks (system identification via OKID example is broken, linked issue has fixes)
- There is a whole 'nother book on this chapter: http://dmdbook.com/ Code too!
- What are the differences among Proper Orthogonal Decomposition (POD), Singular value decomposition (SVD) and principal component analysis (PCA)? - Data Science Stack Exchange

Week 12: Finish Ch 7

Read through end of Ch 7
Meet Sept 12 (link)

Most interesting facts from this section:

- Koopman theory is basically moving the nonlinear part out of control and into the "measurement" subsystem by creating/finding combinations of measurements that let the dynamics be linear.
- Tom mentioned being interested in how this can be applied to the frequency domain: put in the overall frequency response and get back the modes.

Discussion questions:

- Who was in charge of writing this chapter?
- What does "eigen" mean?
    - German for "characteristic"
        - Eigenvalue - scaler
        - Eigenvector - basis
        - Eigensystem - eigenvalue and eigenvector together
    - Eigenfunction

            - Eigenvector where the vector is a function
- What is Hilbert Space?
    - Vector space with an inner product which is a way of measuring angles between vectors. Instead of points, usually it is functions. Orthogonal functions (like sin x, cos x) are nice to use, like orthogonal unit vectors..
    - Sin x and Cos x are functions and they are the basis for Fourier (Fourier space is a part of Hilbert space). This is continuous Fourier: adding a series of sins and coss; this is about the functions, not the FFT.
- Noether and conservations and symmetry is neat
- What are modes?
    - This is still a little unclear

Notes:

- Should we stay here?
    - Koopman doesn't come up again until chapter 10, the next chapter is more about the physics of the inverted pendulum series and adding control to the system.

Week 13: Start Ch 8: LQR to Kalman

Read through end of Ch 8.5 (Read through Kalman)
Meet Sept 19 (link)
Videos for Ch8 are the Controls Bootcamp series which is long (longer than reading the chapter) but a lot more explication (and useful too)

Most interesting facts from this section:

- Eigenvalues and control stability are related (different disciplines in my education).
    - If the eigenvalue is negative, then as time goes to infinity, the system is stable because e^(At). If the eigenvalue is positive then as time goes on, the system doesn't go to zero at infinity.
    - In a transfer function, the eigenvalues are the poles underneath. (Waaaah?!?)

Discussion questions:

- How long can we spend on this section?

Notes:

- Often if you want to optimize something you find where its gradient goes to zero. However, if you want to optimize a constraint equation (something where a solution needs to be satisfied, like optimizing a cost function)

For example, our cost function here is:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}.$$

- Hamiltonian vs Lagrange vs Jacobian vs Riccati what are all these things?
  - Jacobian: (partial) derivative matrix to approximate a function through time
  - Lagrangian multiplier for constraints
    - gradient of one side is proportional to the gradient of the other side
    - the gradient of the thing you want optimize is parallel to the gradient of the constraint
  - Hamiltonian: Apply the math to instead of something to derive the math from?
    - https://en.wikipedia.org/wiki/Hamiltonian_mechanics
    - https://en.wikipedia.org/wiki/Hamiltonian_(control_theory)
  -
- Tom: There is a good podcast with Steve Brunton here, including the motivation for writing the book: Machine Learning For Fluid Mechanics - Steven Brunton | Podcast #50 There is a lightning round at the end, called Question Rampage. This podcast may make this section a bit more clear as it gives more reasoning behind why things like physics are glossed over.
- William Rowan Hamilton (Science YouTuber Collab) | A Capella Science (Chef's Kiss! Fantastic!)

Week 14: Start Ch 8 continued

Read through end of Ch 8? Or as far as you can? We'll be here for another week.
Meet Sept 19 (link)

Videos for Ch8 are the Controls Bootcamp series which is long (longer than reading the chapter) but a lot more explication (and useful too)

Gramian! What a great name. Controllability Gramian The controllability Gramian can be found as the solution of the Lyapunov equation.

Most interesting facts from this section:

- The way the Vs in the Kalman are like no biggie. Seriously, these are impossible. Also, I don't really get it, I thought Kf would change as part of the update step.

Discussion questions:

- How long can we spend on this section?
- Riccati equation: could someone explain this?

Notes:

- Portraits of a Kalman Filter (Elecia's cartoons, not a similar formulation)
- Backlash isn't taken into account in mechanical mathematics, likely to be a problem with a real system. Static friction and binding too.

- University of Washington remove grad program: Two graduate certificates and a cap stone project leads to a master's degree: Professional Master's and Certificate Programs. Two of the certificates:
    - Certificate in Artificial Intelligence and Machine Learning for Engineering
    - Graduate Certificate in Data-Driven Dynamic Systems and Controls for Engineering
- Maxima: A Computer Algebra System
    - Mathematica probably has a better interface (but is more expensive than free)
- I have been trying to update the inverted pendulum colab notebook with more examples from the video lectures: ECW_Extended_CH08_SEC07_1_LQR.ipynb

Week 14: Toward Finishing Ch 8

Read through the end of Ch 8? Or as far as you can? We'll be here for another week.
Meet Oct 3 (link)

Videos for Ch8 are the Controls Bootcamp series which is long (longer than reading the chapter) but a lot more explication (and useful too)

Most interesting facts from this section:
- Actual connection between eigenvalues and poles. Weird that we still need Bode plots.
- Really nice to see the Control Bootcamp: Example Frequency Response (Bode Plot) for Spring-Mass-Damper

Discussion questions:
- Where are you? Video or book?
    - Elecia: Vid27: Laplace Transform and Transfer Function
    - Linda: Cruise Control Example
    - Tom: Vid27: Laplace Transform and Transfer Function
- Convolutional integration: is this just regular integration? Or something else? Is there an integral sign with a circle on it?

Notes:
- Tom, can you send a link to your note sets?
- Guaranteed Margins for LQG Regulators <- the Doyle paper discussed in the videos about stability margins
- Convolutional integrals (this just means convolution, don't be confused by the term… though convolution is an integral)
    - If you have a linear time-independent system, and you know its impulse response, you can use the convolution integral to find out the response to any arbitrary input. To do so, first break the arbitrary input into a bunch of impulses,

shifted in time and amplitude, then sum the impulse responses to get the total response.
  - By taking this to the limit, the impulses get infinitesimally close together in time, and your sum becomes an integral: the convolution integral.
  - The Convolution as A Sum of Impulse Responses
  - Convolution (Wikipedia)
  - These have *nothing* to do with line integrals which have the circle on the integral sign. And really, Line integral convolution is another thing, I think.
- Dynamical Analogies by Olson (Free! 1943) How electrical, mechanical (rotational and linear), and acoustic systems are pretty much all the same. Very cool book!

Week 15: Finish Ch 8

Finish Ch8
Meet Oct 10 (link)

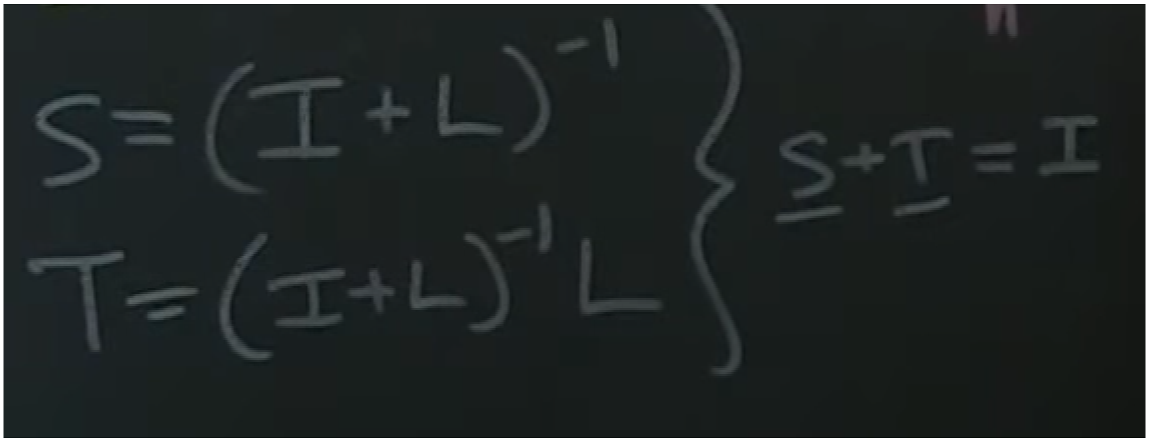
Videos for Ch8 are the Controls Bootcamp series which is long (longer than reading the chapter) but a lot more explication (and useful too)

Most interesting facts from this section:

- Non minimum phase: “goes in the wrong direction first”.
- Parallel parking is provably hard.

Discussion questions:

- This makes me want to understand dynamics better. And the DMD stuff but this chapter it is dynamics. What topics do you want to learn (or learn better)? If you were building a system for Matrix style learning, what would you want to build?

- 

  ### Verification of $\mathbf{S} + \mathbf{T} = \mathbf{I}$

  The relationship you noted, $\mathbf{S} + \mathbf{T} = \mathbf{I}$, is fundamental and can be easily proven using the provided formulas:

  $$\mathbf{S} + \mathbf{T} = (\mathbf{I} + \mathbf{L})^{-1} + (\mathbf{I} + \mathbf{L})^{-1}\mathbf{L}$$

  1. **Factor out** the common term $(\mathbf{I} + \mathbf{L})^{-1}$:

     $$\mathbf{S} + \mathbf{T} = (\mathbf{I} + \mathbf{L})^{-1}(\mathbf{I} + \mathbf{L})$$

     (Note: We use $\mathbf{I}$ as the identity for the first term since we are adding a matrix $\mathbf{L}$ inside the parenthesis.)

  2. **Multiply** the matrix by its inverse:

     $$\mathbf{S} + \mathbf{T} = \mathbf{I}$$

  How does S+T = I? Gemini described it but this is very unintuitive to me.

Notes:
- The controls bookcamp code (matlab only) seems to be here.
- The loop shaping and sensitivity and robustness videos seemed impossible to understand unless you already understood it.
  - Nyquist stability criterion - Wikipedia
  - Multivariable Feedback Control (on author's site)
    - Everything should be PID?

Week 16: 9.1 and most of 9.2

Finish Chapter 9.1, start 9.2
Meet Oct 17 (link)
Lots of videos! They also cover chapter 10

Most interesting facts from this section:

- POD, PCA, and SVD are all dimensionality reduction techniques that are mathematically related, but differ in their application and context. **SVD is a general matrix factorization method, while PCA uses the SVD (or eigen-decomposition) to find the principal components (axes of maximum variance) in a statistical dataset**. POD is a method often used in physics and engineering that is essentially an application of SVD to identify the dominant, energy-carrying modes in a physical system, like fluid flow
- POD is the physics version of PCA which is the statistic version of SVD
  - This sorts the eigenvalues with the most energy (most important features)
    - Does matlab do this sorting or is it a function of SVD itself?
- SVD is the underlying algorithm and attempts to diagonalize a matrix that might not be square or otherwise mathematically nice. SVD is finding the eigenvectors.
- Matrices within vectors, matrices within matrices. Block matrices (aka partitioning).

Discussion questions:

- Describe POD: what was this chapter about?
- Matrix manipulation: internals (Vid 8)

Notes:

- Moore 1981 paper: Principal component analysis in linear systems: Controllability, observability, and model reduction | IEEE Journals & Magazine (free download here)
- Got CH09_SEC02_1_GramianPlot.ipynb colab to work with some modifications, committed to my fork of databook_python.
- Green's theorem: integrate a function around an edge of curve out to be related to the derivative of the interior (volume) of the curve in a specific (magical?) way
- Lyaponov made another appearance in 9.2.
  - If you're into Lyapunov stability, in the Haykin Neural Networks textbook they analyze recurrent NNs as a dynamical system with Lyapunov's direct method. This is in chapter 13, Neurodynamics.
  -
  - URL: https://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf
  - Lyaponov equation.
  - Controllability Gramian - Wikipedia: "The controllability Gramian can be found as the solution of the Lyapunov equation"
- Block matrix (Linda will look up something to better explain them)

Week 17: 9.3 (Finish 9)

Finish Chapter 9.1, start 9.2
Meet Oct 31 (link)
Video: Stop when you get vid 20.

Most interesting facts from this section:

- Make a system identification: OKID to stop needing an impulse for ERA. This is black magic. But how well does it work? How random is pseudo-random?
- Do not sit on shake tables because random vibrations may destroy your kidneys.

Discussion questions:

- What data would be interesting to do this to? What is stopping you?

Notes:

Running CH09_SEC03_ERA_OKID.ipynb in collab, I needed to prefix it with

```
import os
os.environ['BLA_VENDOR']="Generic"

!pip install control
!pip install slycot
```

Updated in my repo to run in collab
https://github.com/eleciawhite/databook_python/blob/master/CH09/CH09_SEC03_ERA_OKID.ipynb

https://gemini.google.com/share/3cf2ffc7cd35

Week 18: Read through 10.2 (mid)

Meet Nov 14 (link)
Video play list is “Data-Driven Control with Machine Learning”
Start at vid 20, go to vid 32

Most interesting facts from this section:

- [Lotka–Volterra equations - Wikipedia] (bunnies and wolves)

Discussion questions:

- Jumped the shark? Complicated on complicated. Is it usable? The next section is less interpretable. How important is that?
- How do you stay engaged with highly technical material? How do you come back to difficult material?
- Coloring books (botany, periodic table, standard drugs). What would you put in an infocomic or coloring book for this book?
- Genetic algorithms? So 1980s!

Notes:

(This really goes back to an older chapter but control was added to some this time so I summarized it here.)

System ID Tool	**System type**	**Description**
POD: Proper orthogonal decomposition	Linear	Start with high dimensional snapshots of the system. Essentially PCA or SVD: find eigenvalues, use those to represent the system. Reduce state dimensions (not really system ID)
DMD: Dynamic Mode Decomp	Linear	Start with high dimensional snapshots of the system. Find modes (spatial) and their eigenvalues (temporal, frequency). Fastest to compute.
Koopman Analysis	Linear representation of nonlinear system	Add different (non linear) functions of the taken measurements (observables). Do DMD.

		Not interpretable (things done in eigenvector space). End up with Koopman mode, eigenvalues, eigenvectors.
SINDy: Sparse Identification of Nonlinear Dynamics	Nonlinear (but sparse) representation of nonlinear system	Match a library of equation terms (beyond the measurements). Very interpretable.

The Manga Guide to Statistics
The Manga Guide to Linear Algebra
Top 200 Drugs Made Easy: Pharmacology Coloring Book

Week N-1: Almost the end

Finish the book!

Extremum seeking control is cool… but why? It is a nice optimization/control method but it seems extraneous…. And very late in the book. It really depends on creating a well behaved cost function. It also reminded me of ADC dithering. Oh, and there is hardware to do this (400-W GaN-Based MPPT Charge Controller and Power Optimizer Reference Design).

There are Problem Sets | DATA DRIVEN SCIENCE & ENGINEERING but they stop at/before section 3.

What are you interested in learning next as fancy math anonymous?

- Linda: stats and control
    - Papers
    - Toward a theory of perspective perception in pictures | JOV | ARVO Journals
- Tom: Graphics (both GPU math and art visualization: together becomes animation), how does human vision work? Cartoon physics. See Disney's 12 Principles Of Animation: Bringing Characters To Life and the cube video
    - Journal of Vision
- Elecia: machine learning (stats!); origami and math; robotics, kinematic and trig (geometry probably)

Mathematical Methods for Computer Vision, Robotics, and Graphics

Projective geometry - Wikipedia Is “applied projective geometry” is this is a book?

- Projective Geometry: From Foundations to Applications Just math though, not applications
- https://archive.org/details/principlesofproj00hattrich/page/18/mode/2up Even more math and tough writing (1913)
- Geometry and the Imagination by David Hilbert hilbert-geometry-and-the-imagination-configurations.pdf
- Cinderella (Documentation : Theoretical Background)