Introduction & Linear Algebra Basics
What do I need to know?

- I won’t assume you know machine learning or signal processing
  - You are here to learn, not to know!

- Be comfortable with the basics
  - Some linear algebra, some probability
    - If you are rusty that’s ok, you are here to learn
What is this?

- Is it a signals class?
- Is it a machine learning class?
Signal Processing

• The study of capturing, processing and manipulating “signals”

• What is a signal?
Signals (as far as this class goes)

• “Structured” collections of measurements that convey information

Images

Sounds

Brains!!

Stocks
Machine Learning

• The study of discovering and extracting information from “data”

• For data we will use signals
Why bother?

- Traditional signal processing doesn’t really care about its input’s content
- Traditional machine learning is not signals-friendly
MLSP

- MLSP combines both disciplines to perform learning on signals data
- Many examples of MLSP in the real-world
Face recognition

• Found in cameras and photo software
Speech recognition

so lastly i want to see it to show you innovation in natural input as well and
Surveillance

- Detection of specified objects / activity

*Gunshot detection*

*Pedestrian detection*
Bio-signals

- Interpreting our body’s data
Many other applications

- Machine condition monitoring
- Biometrics
- Music Information Retrieval
- Robotics
- Gesture-based UIs
- Network Traffic Analysis
- Financial data mining
- ...
About this course

• Heavy on practical applications
  • Please bring your own domain problem in class!

• Won’t go excruciatingly deep on theory
  • We’ll skip convergence proofs, etc.
    • Many more courses here that cover all that

• Our objective is real-world experience
Syllabus: the basics

• Covering the basics:
  • Part 1: Linear Algebra and Probability
  • Part 2: Signals Theory
  • Part 3: Representations and Features
Syllabus: machine learning review

• Elements of machine learning
  • Part 4: Unsupervised learning basics
  • Part 5: Detection and classification
  • Part 6: Time-series and dynamical models
Syllabus: The fun stuff

- Applications and theory:
  - ICA, MIMO models, Arrays, Sensor Fusion
  - Matrix Factorizations, Bag Models
  - Manifolds and Embedding
  - Graphical Models
  - Compressive Sensing and Sparsity
  - Computer Vision, Speech Recognition
  - Music/Audio Informatics
  - Bio/Brain-signals
  - ...
Your part

• Problem sets: ~40% of the grade
  • One every two weeks, will be mostly machine problems

• Final Project: ~50% of the grade
  • Mid-semester: Proposal due
  • Last 1-2 weeks: Presentations and/or posters

• Remaining 10% of grade
  • Show your face in class, ask questions, make sure I know who you are!
The final project is “conference style”

- Teams of 2-3 students (no more than 3, no less than 2)
  - Make friends now!

- Mid-term: Abstract submission
  - Short abstract describing the problem you want to solve and how you plan to

- Week ~13: Paper Submission
  - 4-6 page paper
  - Peer reviewed by all of you
  - All papers accepted!
Web stuff

• We have a course page:
  • http://courses.engr.illinois.edu/cs598ps
  • Will have lectures, problem sets, data, links, etc.

• We have a piazza.com page
  • Look for CS 598 PS / CS 598 PSO
  • Use for discussions, finding project-mates, etc.
Who am I?

- Instructor: Paris Smaragdis (CS & ECE)
  - paris@illinois.edu
  - Office: Siebel Center 3231
    - Send me email of you want to meet

- Interested in machine perception, computational audition

- Past chair of the IEEE MLSP Technical Committee

- Plenty of commercialized experience on the subject
Who is the TA?

- Minje Kim (CS)
  - minje@illinois.edu

- Office hours at Siebel 0207
  - Mondays 10:00-11:00
  - Online students:
    - Wednesdays 14:30-15:30
Who are you?

- Name, department, advisor, domain interests?
Final administrative note

• This class is oversubscribed
  • If you don’t think you will stay until the end please consider dropping the course so that waitlist students can register

• If you are not formally registered yet, send me your UID
  • We need this for the email list for various announcements
Linear algebra refresher

• Linear algebra is the math tool du jour
  • Compact notation
  • Convenient set operations
  • Used in all modern texts
  • Interfaces well with MATLAB, numpy, R, etc.

• We will use a lot of it!!
Scalars, Vectors, Matrices, Tensors

**Scalars**

\[ \mathbf{x} \]

**Vectors**

\[ \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \]

**Matrix**

\[ \mathbf{X} = \begin{bmatrix} x_1 & \cdots & x_N \\ x_{1,1} & x_{N,1} \\ \vdots & \vdots & \vdots \\ x_{1,M} & x_{N,M} \end{bmatrix} \]

**Tensor**

\[ \mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_K\} = \begin{bmatrix} \mathbf{x}_{1,1,1} & \mathbf{x}_{N,1,1} \\ \vdots & \vdots \\ \mathbf{x}_{1,M,1} & \mathbf{x}_{N,M,1} \\ \vdots & \vdots \\ \mathbf{x}_{1,K,M} & \mathbf{x}_{N,K,M} \end{bmatrix} \]
How will we see these?

- 1D signals (e.g. sounds) will be vectors

\[ x^T = \begin{bmatrix} x(0) & \cdots & x(T) \end{bmatrix} = \begin{bmatrix} \end{bmatrix} \]

- 2D signals (e.g. images) will be matrices

\[ X = \begin{bmatrix} x_{1,1} & x_{N,1} \\ \vdots & \ddots & \vdots \\ x_{1,M} & x_{N,M} \end{bmatrix} = \begin{bmatrix} \end{bmatrix} \]

- 3D data will be videos, etc ...
Element-wise operations

- Addition/subtraction

\[ a \pm b = c \rightarrow a_i \pm b_i = c_i \]

- Multiplication (Hadamard product)

\[ a \odot b = c \rightarrow a_i \cdot b_i = c_i \]

- Other times denoted as \( a \circ b \) or \( a \cdot * b \)

- No named operator for element-wise division
  - Just use Hadamard with inverted elements
Transpose

- Transpose
  - Change rows to columns (and vice versa)

\[
x = \begin{bmatrix}
  x_1 \\
  \vdots \\
  x_N
\end{bmatrix}, \quad x^\top = \begin{bmatrix}
  x_1 & \cdots & x_N
\end{bmatrix}
\]

\[
x = \begin{bmatrix}
  x_1 & x_3 \\
  x_2 & x_4
\end{bmatrix}, \quad x^\top = \begin{bmatrix}
  x_1 & x_2 \\
  x_3 & x_4
\end{bmatrix}
\]

- Hermitian (conjugate transpose)
  - Notated as \(X^H\)
    - MATLAB note: \(x'\) is Hermitian transpose, \(x.\)'. is transpose
Visualizing transposition

- Mostly pointless for 1D signals

\[ x = \begin{bmatrix} \text{1D signal} \end{bmatrix}, \quad x^\top = \begin{bmatrix} \text{1D signal} \end{bmatrix} \]

- Swap dimensions for 2D signals

\[ x = \begin{bmatrix} \text{2D image} \end{bmatrix}, \quad x^\top = \begin{bmatrix} \text{2D image} \end{bmatrix} \]
Reshaping operators

• The vec operator
  • Unrolls elements column-wise
  • Useful for getting rid of matrices/tensors

\[
\text{vec}(\mathbf{x}) = \text{vec} \begin{bmatrix} x_1 & x_3 \\ x_2 & x_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
\]

• The vec-transpose
  • For \((p)\) of an \(m \times n\) matrix make each column a \(p \times (m/p)\) matrix
  • Useful for inverting vec and getting rid of tensors \(\mathbf{X} = \text{vec}(\mathbf{X})^{(M)}, \mathbf{X} \in \mathbb{R}^{M \times N}\)

\[
\mathbf{x}^{(2)} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \\ x_{41} & x_{42} \\ x_{51} & x_{52} \\ x_{61} & x_{62} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{31} & x_{51} \\ x_{21} & x_{41} & x_{61} \\ x_{12} & x_{32} & x_{52} \\ x_{22} & x_{42} & x_{62} \end{bmatrix}
\]
Trace and diag

- **Matrix trace**
  - Sum of diagonal elements
    \[
    \text{tr}(X) = \text{tr} \begin{bmatrix}
    x_{11} & \cdots & x_{1N} \\
    \vdots & \ddots & \vdots \\
    x_{N1} & \cdots & x_{NN}
    \end{bmatrix} = \sum_i x_{ii}
    \]

- **The diag operator**
  \[
  \text{diag}(x) = \text{diag} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 & 0 \\ 0 & x_2 \end{bmatrix}
  \]
  \[
  \text{diag}^{-1} \begin{bmatrix} x_1 & a \\ b & x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
  \]
The dot product

- **Dot product**
  - Shorthand for multiply and accumulate
  
  $$\mathbf{x}^\top \cdot \mathbf{y} = \sum_i x_i \cdot y_i = |\mathbf{x}| \cdot |\mathbf{y}| \cos \theta$$

- **Geometry**
  - For unit vectors:
    
    $$\theta = \arccos (\mathbf{x}^\top \cdot \mathbf{y})$$

  - Great tool for checking out similarity
Matrix-vector product

• Generalizing the dot product

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 \\
\end{bmatrix} \cdot y = \begin{bmatrix}
  x_1 \cdot y \\
  x_2 \cdot y \\
  x_3 \cdot y \\
\end{bmatrix} = \begin{bmatrix}
  \sum x_{1,i} \cdot y_i \\
  \sum x_{2,i} \cdot y_i \\
  \sum x_{3,i} \cdot y_i \\
\end{bmatrix}
\]

• \( x \) must have as many columns as \( y \) has elements
  • Non-commutative!

• Useful for computing multiple dot products
  • Pack all vectors that you want to multiply in a matrix
Matrix-matrix product

• Between two matrices:

\[
X \cdot Y = \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
\end{bmatrix} \cdot \begin{bmatrix}
y_1 & y_2 & y_3 \\
\end{bmatrix} = \begin{bmatrix}
x_1 \cdot y_1 & x_1 \cdot y_2 & x_1 \cdot y_3 \\
x_2 \cdot y_1 & x_2 \cdot y_2 & x_2 \cdot y_3 \\
x_3 \cdot y_1 & x_3 \cdot y_2 & x_3 \cdot y_3 \\
\end{bmatrix}
\]

• \(X\) must have as many columns as \(Y\) has rows
  • \((M \times N) = (M \times K) \cdot (K \times N)\)
  • Remember it doesn’t commute!

• All linear operations can be represented as a matrix product
  • We’ll be seeing that a lot!
Matrix products

- Output rows == left matrix rows
- Output columns == right matrix columns
Visualizing the matrix product
Visualizing the matrix product

\[ \text{Image} = \text{Matrix} \times \text{Image} \]
Visualizing the matrix product

\[
\begin{bmatrix}
0 & 1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}
= 
\begin{bmatrix}
0.5 & 0.5 & 0.5 \\
0.5 & 0.5 & 0.5 \\
0.5 & 0.5 & 0.5
\end{bmatrix}
\]
Visualizing the matrix product

\[ \begin{align*}
\text{matrix} & \quad \text{vector} \quad \text{vector} \\
\times & \quad = \quad = \quad =
\end{align*} \]
Visualizing the matrix product
Norms

- 2-norm:
  \[\|x\| = \sqrt{\sum x_i^2}\]

- \(p\)-norms:
  \[\|x\| = \left(\sum |x_i|^p\right)^{1/p}\]

- Frobenius norm:
  \[\|X\|_F = \sqrt{\text{tr}(X^\top \cdot X)} = \sqrt{\text{tr}\left(\begin{bmatrix} x_1 & x_2 \\ x_1 & x_2 \end{bmatrix}\right)} = \sqrt{\text{tr}\left(\begin{bmatrix} x_1 \cdot x_1 & x_1 \cdot x_2 \\ x_2 \cdot x_1 & x_2 \cdot x_2 \end{bmatrix}\right)}\]
Kronecker product

- A bit more complex
  - Replicate and multiply right matrix with each scalar of left matrix

\[
\begin{bmatrix}
  x_{11} & x_{12} \\
  x_{21} & x_{22}
\end{bmatrix}
\otimes Y =
\begin{bmatrix}
  x_{11} \cdot Y & x_{12} \cdot Y \\
  x_{21} \cdot Y & x_{22} \cdot Y
\end{bmatrix}
\]

- Useful result:
  \[
  \text{vec}(X \cdot Y \cdot Z) = (Z^\top \otimes X) \text{vec}(Y)
  \]
Visualizing Kronecker

\[
\begin{bmatrix}
1 & 3 \\
2 & 0 \\
\end{bmatrix} \times \ = \ ?
\]
Visualizing Kronecker

\[
\begin{bmatrix}
1 & 3 \\
2 & 0
\end{bmatrix} \times =
\]

\[
\begin{bmatrix}
\begin{bmatrix}
\end{bmatrix}
\end{bmatrix}
\]
Dealing with tensors

• Using Kronecker products and the vec operator we can perform multilinear transforms
• Tensor example with RGB images:
Mixing the colors

- Define color, horizontal and vertical mixing

\[
\left( C^T \otimes H^T \otimes V^T \right) \cdot \text{vec}(X)
\]

- Example: color mixing

\[
\left( \text{diag}\begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \otimes I \otimes I \right) \text{vec}(X) =
\]

- Caution: Dimensions will quickly get out of hand this way
Special matrices

- **Symmetric:**
  \[ X = X^\top \implies x_{ij} = x_{ji} \]

- **Positive definite**
  - Is so if: \( y^\top \cdot X \cdot y > 0, \ \forall \ y \)
  - Also symmetric

- **Orthonormal:**
  \[ X^\top \cdot X = X \cdot X^\top = I \]
Matrix inverse

• “Undoes” a matrix multiplication
  • Only for square matrices
  • Not all matrices are invertible
    • must be a full-rank matrix

\[ X^{-1} \cdot X = I \]
\[ X^{-1} \cdot X \cdot Y = Y \]
\[ Y \cdot X \cdot X^{-1} = Y \]

• Remember, in matrix multiplication order matters

\[ X \cdot Y \cdot X^{-1} \neq Y \]
Matrix pseudoinverse

- Also known as Moore-Penrose (or MP) pseudoinverse
  - For an $m \times n$ matrix $X$, pseudoinverse is $n \times m$ matrix $X^+$
    
    \[
    X \cdot X^+ \cdot X = X \\
    X^+ \cdot X \cdot X^+ = X^+ \\
    (X \cdot X^+)^T = X \cdot X^+ \\
    (X^+ \cdot X)^T = X^+ \cdot X
    \]
  - We’ll be seeing this operation a lot, it’s essentially least squares
    
    $$A \cdot x = y \rightarrow x = A^+ \cdot y$$
Eigenanalysis

- Eigenvectors and eigenvalues
  - For an $n \times n$ matrix $X$
    \[
    X \cdot V = V \cdot L
    \]
    \[
    V = \begin{bmatrix} v_1 & \cdots & v_N \end{bmatrix}
    \]
    \[
    L = \text{diag} \begin{bmatrix} \lambda_1 & \cdots & \lambda_N \end{bmatrix}
    \]
  - $V$ is $m \times n$ and contains the eigenvectors $v_i$
    - It will be an orthogonal matrix for positive (semi-)definite matrix inputs
  - $L$ is $n \times n$ contains the eigenvalues $\lambda_i$
Low rank approximations

• Use smaller matrices to describe a large matrix

\[ X \approx A \cdot B \]

• With \( X \) being \( m \times n \), \( A \) being \( m \times r \), \( B \) being \( r \times n \), and \( r < m \)
The Singular Value Decomposition (SVD)

- Similar decomposition to eigenanalysis
  - For a matrix $X$
    \[ X = U \cdot S \cdot V^\top \]
    \[ U^\top \cdot U = I \]
    \[ V^\top \cdot V = I \]
    \[ S = \text{diag}(\sigma_i) \]
- $U$, $V$ are orthonormal
  - Contain the left and right singular vectors of $X$
- $S$ is diagonal
  - Contains on its diagonal the singular values $\sigma_i$
Visualizing the SVD

- Comes in two versions, full and economy
  - The only difference is the size of the matrices
    - The numerical approximation is the same
  - By truncating the columns of $S$ we can make a low-rank approx.
Recap

• What’s this class about?
  • Signals, learning, fun, etc ...

• Linear algebra basics
  • Algebraic operations, norms, decompositions, form manipulations
Finale

• Skim through this material for now
  • We’ll be seeing it in context soon
    • e.g., what are the eigenvectors of image matrices?

• Reading material
  • Old and new algebra useful for statistics
  • The Matrix Cookbook
    • http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=3274
Questions?

• Slides will be online soon after all lectures at:
  • http://courses.engr.illinois.edu/cs598ps/
Next lecture

• Review of:
  • Probability theory
    • Bayes theorem, probability rules, Bayes nets
    • Basic distributions, transformations of RV’s
  • Statistics
    • Basic measures, independence, information
  • Parameter estimation intro
    • Maximum likelihood, MAP, Bayesian, EM intro