

10-606 Mathematical Foundations for Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University



Course Overview

Matt Gormley
Lecture 1
August 27, 2018

ABOUT THIS COURSE

How to describe 606/607 to a friend

606/607 is...

a **formal** presentation of **mathematics** and **computer science**...

motivated by (carefully chosen) **real-world problems** that arise in **machine learning**...

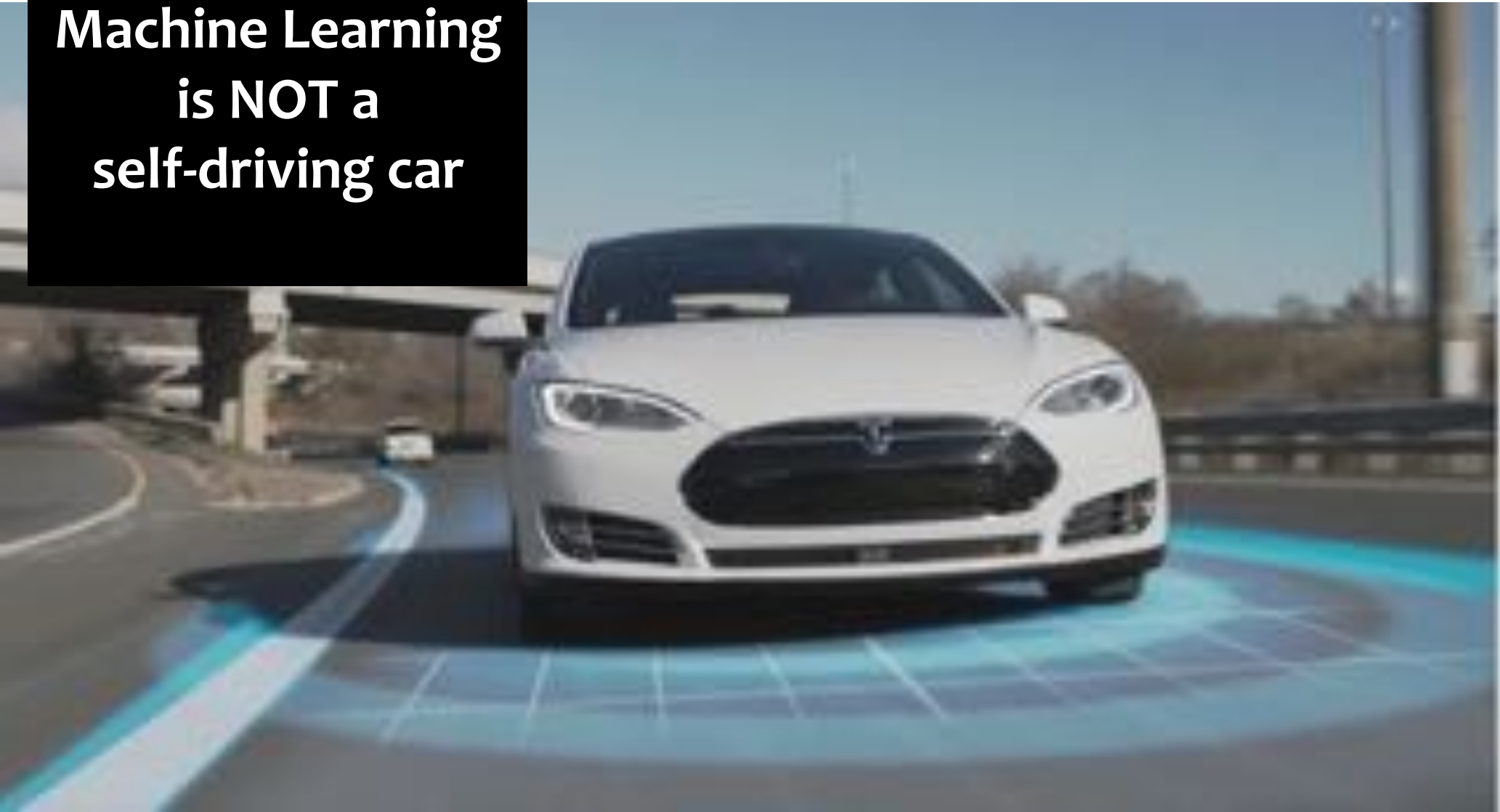
where the **broader picture** of how those problems arise is treated **somewhat informally**.

How to describe 606/607 to a friend

606/607 is...

The class you should take to
prepare for future coursework
in machine learning

**Machine Learning
is NOT a
self-driving car**



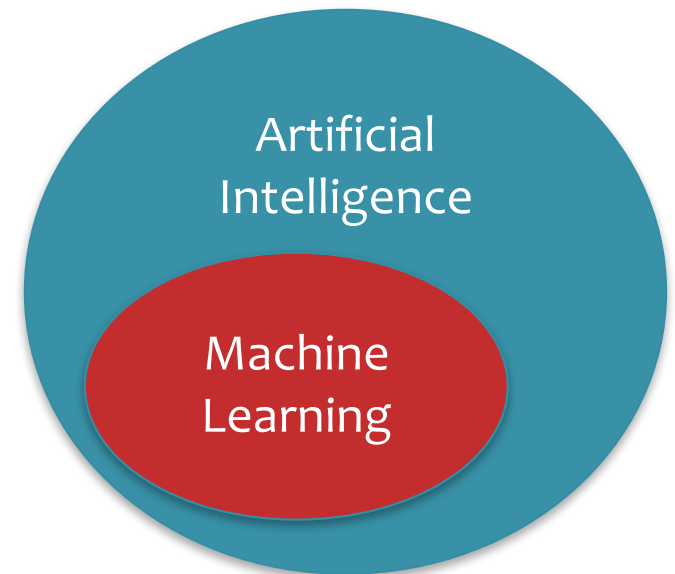
WHAT IS MACHINE LEARNING?

Artificial Intelligence

The basic goal of AI is to develop intelligent machines.

This consists of many sub-goals:

- Perception
- Reasoning
- Control / Motion / Manipulation
- Planning
- Communication
- Creativity
- Learning



What is Machine Learning?

The goal of this course is to provide you with a toolbox:

Machine Learning

Statistics

Probability

Computer Science

Optimization



What is ML?

Computer
Science

Domain of
Interest

Machine Learning

Optimization

Statistics

Probability

Calculus

Measure
Theory

Linear Algebra

What is ML?

Computer
Science

Domain of
Interest

Machine Learning

Optimization

Statistics

Probability

Calculus

Measure
Theory

Linear Algebra

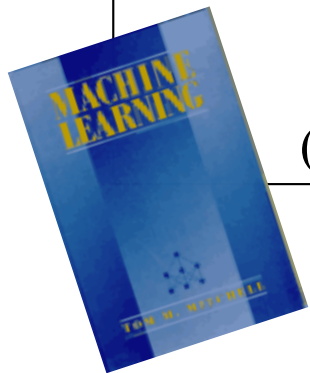
Speech Recognition

1. Learning to recognize spoken words

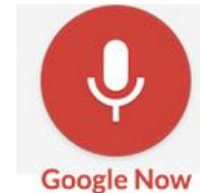
THEN

“...the SPHINX system (e.g. Lee 1989) learns speaker-specific strategies for recognizing the primitive sounds (phonemes) and words from the observed speech signal...neural network methods...hidden Markov models...”

(Mitchell, 1997)



NOW



Source: <https://www.stonetemple.com/great-knowledge-box-showdown/#VoiceStudyResults>

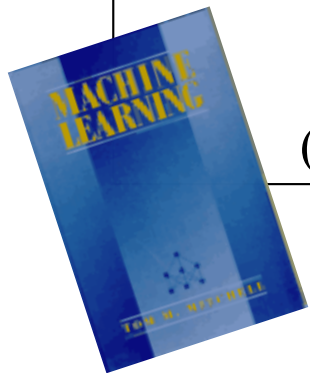
Robotics

2. Learning to drive an autonomous vehicle

THEN

“...the ALVINN system (Pomerleau 1989) has used its learned strategies to drive unassisted at 70 miles per hour for 90 miles on public highways among other cars...”

(Mitchell, 1997)



NOW



waymo.com

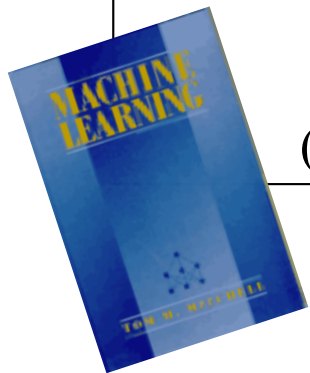
Robotics

2. Learning to drive an autonomous vehicle

THEN

“...the ALVINN system (Pomerleau 1989) has used its learned strategies to drive unassisted at 70 miles per hour for 90 miles on public highways among other cars...”

(Mitchell, 1997)



NOW



<https://www.geek.com/wp-content/uploads/2016/03/uber.jpg>

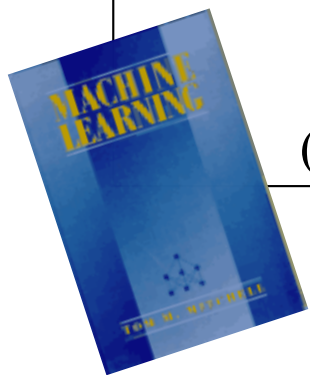
Games / Reasoning

3. Learning to beat the masters at board games

THEN

“...the world’s top computer program for backgammon, TD-GAMMON (Tesauro, 1992, 1995), learned its strategy by playing over one million practice games against itself...”

(Mitchell, 1997)



NOW

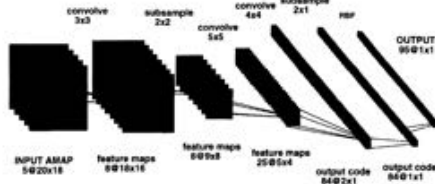


Computer Vision

4. Learning to recognize images

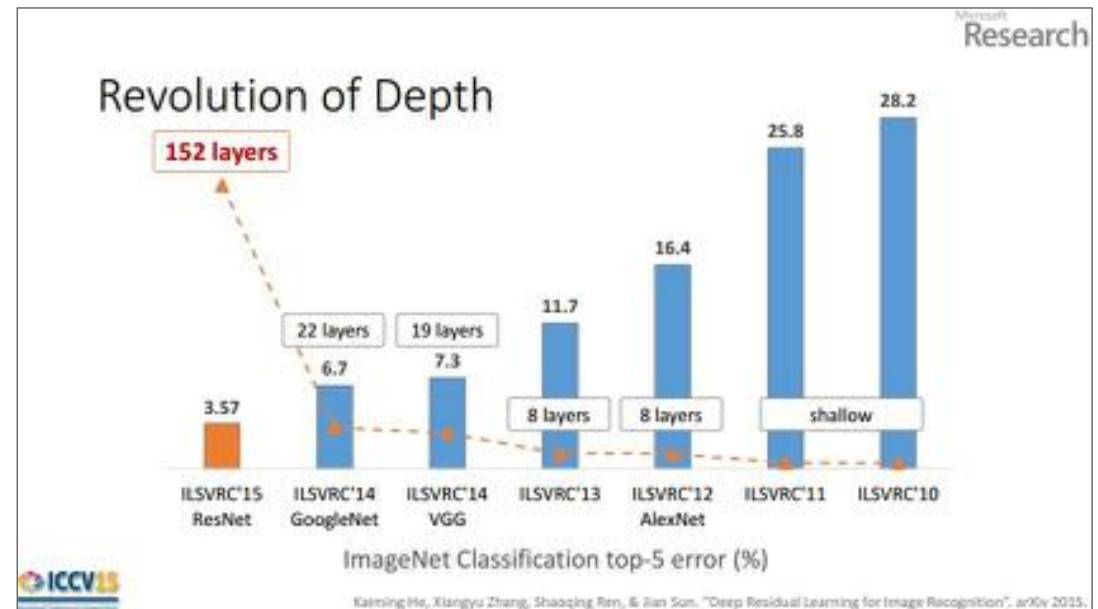
THEN

“...The recognizer is a convolution network that can be spatially replicated. From the network output, a hidden Markov model produces word scores. The entire system is globally trained to minimize word-level errors....”



(LeCun et al., 1995)

NOW



Learning Theory

• 5. In what cases and how well can we learn?

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about...

	Realizable	Agnostic
Finite $ \mathcal{H} $	$N \geq \frac{1}{\epsilon} [\log(\mathcal{H}) + \log(\frac{1}{\delta})]$ labeled examples are sufficient so that with probability $(1 - \delta)$ all $h \in \mathcal{H}$ with $R(h) \geq \epsilon$ have $\hat{R}(h) > 0$.	$N \geq \frac{1}{2\epsilon^2} [\log(\mathcal{H}) + \log(\frac{2}{\delta})]$ labeled examples are sufficient so that with probability $(1 - \delta)$ for all $h \in \mathcal{H}$ we have that $ R(h) - \hat{R}(h) < \epsilon$.
Infinite $ \mathcal{H} $	$N = O(\frac{1}{\epsilon} [\text{VC}(\mathcal{H}) \log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta})])$ labeled examples are sufficient so that with probability $(1 - \delta)$ all $h \in \mathcal{H}$ with $R(h) \geq \epsilon$ have $\hat{R}(h) > 0$.	$N = O(\frac{1}{\epsilon^2} [\text{VC}(\mathcal{H}) + \log(\frac{1}{\delta})])$ labeled examples are sufficient so that with probability $(1 - \delta)$ for all $h \in \mathcal{H}$ we have that $ R(h) - \hat{R}(h) \leq \epsilon$.

Two Types of Error

① True Error (aka. expected risk) (aka. Generalization Error)

$$R(h) = \mathbb{P}_{x \sim p^*(x)} (c^*(x) \neq h(x)) \quad \leftarrow \text{always unknown.}$$

② Train Error (aka. empirical risk)

$$\begin{aligned} \hat{R}(h) &= \mathbb{P}_{x \sim S} (c^*(x) \neq h(x)) \quad \leftarrow S = \{x^{(1)}, \dots, x^{(N)}\} \\ &= \frac{1}{N} \sum_{i=1}^N \mathbb{1}(c^*(x^{(i)}) \neq h(x^{(i)})) \quad \leftarrow \text{known, computable} \\ &= \frac{1}{N} \sum_{i=1}^N \mathbb{1}(y^{(i)} \neq h(x^{(i)})) \end{aligned}$$

PAC Learning

Q: Can we bound $R(h)$ in terms of $\hat{R}(h)$?
A: Yes!

PAC stands for Probably Approximately Correct

PAC learner yields hypothesis h , which is approximately correct $R(h) \approx 0$
with high probability $\Pr(R(h) \approx 0) \approx 1$

Def = PAC Criterion

$$\Pr(\forall h, |R(h) - \hat{R}(h)| \leq \epsilon) \geq 1 - \delta$$

1. How many examples do we need to learn?
2. How do we quantify our ability to generalize to unseen data?
3. Which algorithms are better suited to specific learning settings?

What is Machine Learning?

The goal of this course is to provide you with a toolbox:

Machine Learning

Statistics

Probability

Computer Science

Optimization

To solve all the problems above and more



Topics Covered in an ML Course

(e.g. 10-601)

- Foundations
 - Probability
 - MLE, MAP
 - Optimization
- Classifiers
 - KNN
 - Naïve Bayes
 - Logistic Regression
 - Perceptron
 - SVM
- Regression
 - Linear Regression
- Important Concepts
 - Kernels
 - Regularization and Overfitting
 - Experimental Design
- Unsupervised Learning
 - K-means / Lloyd's method
 - PCA
 - EM / GMMs
- Neural Networks
 - Feedforward Neural Nets
 - Basic architectures
 - Backpropagation
 - CNNs
- Graphical Models
 - Bayesian Networks
 - HMMs
 - Learning and Inference
- Learning Theory
 - Statistical Estimation (covered right before midterm)
 - PAC Learning
- Other Learning Paradigms
 - Matrix Factorization
 - Reinforcement Learning
 - Information Theory

ML Big Picture

Learning Paradigms:

What data is available and when? What form of prediction?

- supervised learning
- unsupervised learning
- semi-supervised learning
- reinforcement learning
- active learning
- imitation learning
- domain adaptation
- online learning
- density estimation
- recommender systems
- feature learning
- manifold learning
- dimensionality reduction
- ensemble learning
- distant supervision
- hyperparameter optimization

Theoretical Foundations:

What principles guide learning?

- ☐ probabilistic
- ☐ information theoretic
- ☐ evolutionary search
- ☐ ML as optimization

Problem Formulation:

What is the structure of our output prediction?

boolean	Binary Classification
categorical	Multiclass Classification
ordinal	Ordinal Classification
real	Regression
ordering	Ranking
multiple discrete	Structured Prediction
multiple continuous	(e.g. dynamical systems)
both discrete & cont.	(e.g. mixed graphical models)

Facets of Building ML Systems:

How to build systems that are robust, efficient, adaptive, effective?

1. Data prep
2. Model selection
3. Training (optimization / search)
4. Hyperparameter tuning on validation data
5. (Blind) Assessment on test data

Big Ideas in ML:

Which are the ideas driving development of the field?

- inductive bias
- generalization / overfitting
- bias-variance decomposition
- generative vs. discriminative
- deep nets, graphical models
- PAC learning
- distant rewards

Application Areas

Key challenges?

NLP, Speech, Computer Vision, Robotics, Medicine, Search

**WHY DO WE NEED MATH / CS FOR
MACHINE LEARNING?**

Why Math for ML?

To best understand A we need B

A

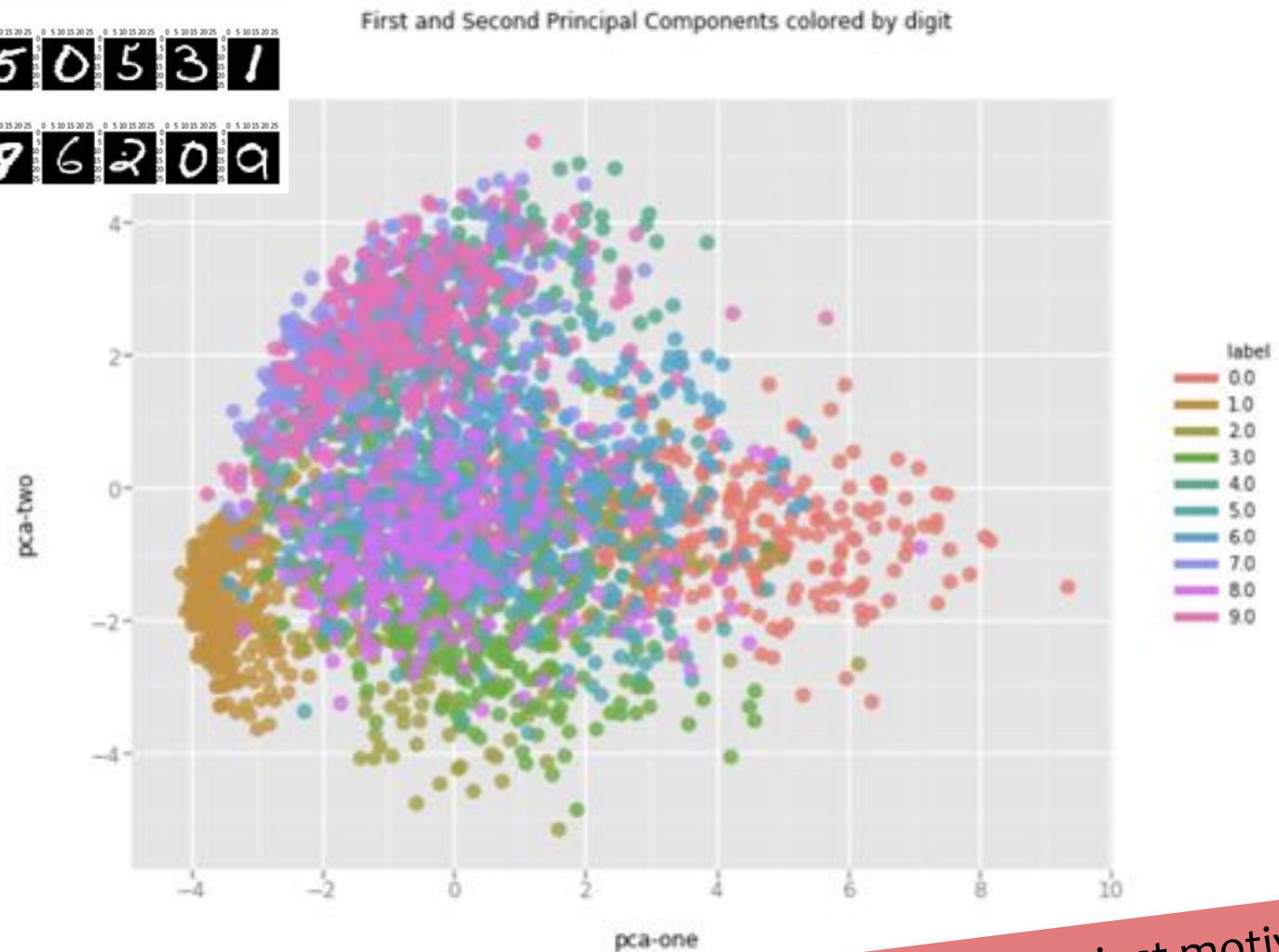
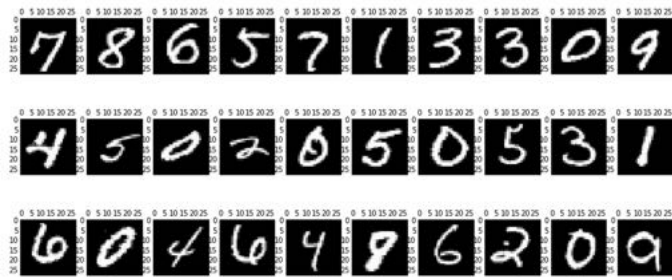
B

Why Math for ML?

To best understand A we need B

A	B
Derivation of Principal Component Analysis (PCA)	Linear Algebra <ul style="list-style-type: none">• Vector spaces, Functions and Function Spaces• Matrices and linear operators• Matrix decomposition

Principle Component Analysis (PCA)



Note: This is just motivation – we'll cover the math need to understand these topics later!

SVD for PCA

For any arbitrary matrix \mathbf{A} , SVD gives a decomposition:

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \quad (1)$$

where $\mathbf{\Lambda}$ is a diagonal matrix, and \mathbf{U} and \mathbf{V} are orthogonal matrices.

Suppose we obtain an SVD of our data matrix \mathbf{X} , so that:

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \quad (1)$$

Now consider what happens when we rewrite $\mathbf{\Sigma} = \frac{1}{N}\mathbf{X}^T\mathbf{X}$ terms of this SVD.

$$\mathbf{\Sigma} = \frac{1}{N}\mathbf{X}^T\mathbf{X} \quad (2)$$

$$= \frac{1}{N}(\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T)^T(\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T) \quad (3)$$

$$= \frac{1}{N}(\mathbf{V}\mathbf{\Lambda}^T\mathbf{U}^T)(\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T) \quad (4)$$

$$= \frac{1}{N}\mathbf{V}\mathbf{\Lambda}^T\mathbf{\Lambda}\mathbf{V}^T \quad (5)$$

$$= \frac{1}{N}\mathbf{V}(\mathbf{\Lambda})^2\mathbf{V}^T \quad (6)$$

Above we used the fact that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ since \mathbf{U} is orthogonal by definition.

Note: This is just motivation – we'll cover the math need to understand these topics later!

SVD for PCA

For any arbitrary matrix \mathbf{A} , SVD gives a decomposition:

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \quad (1)$$

where $\mathbf{\Lambda}$ is a diagonal matrix

Suppose we obtain the SVD of \mathbf{X}

Now consider what happens if we run SVD on $\mathbf{X}^T\mathbf{X}$.
of this SVD.

We find that $(\mathbf{\Lambda})^2$ is a diagonal matrix whose entries are $\Lambda_{ii} = \lambda_i^2$ the squares of the eigenvalues of the SVD of \mathbf{X} . Further, both \mathbf{X} and $\mathbf{X}^T\mathbf{X}$ share the same eigenvectors in their SVD.

Thus, we can run SVD on \mathbf{X} without ever instantiating the large $\mathbf{X}^T\mathbf{X}$ to obtain the necessary principal components more efficiently.

$$\mathbf{\Sigma} = \frac{1}{N}\mathbf{X}^T\mathbf{X} \quad (2)$$

$$= \frac{1}{N}(\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T)^T(\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T) \quad (3)$$

$$= \frac{1}{N}(\mathbf{V}\mathbf{\Lambda}^T\mathbf{U}^T)(\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T) \quad (4)$$

$$= \frac{1}{N}\mathbf{V}\mathbf{\Lambda}^T\mathbf{\Lambda}\mathbf{V}^T \quad (5)$$

$$= \frac{1}{N}\mathbf{V}(\mathbf{\Lambda})^2\mathbf{V}^T \quad (6)$$

Above we used the fact that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ since \mathbf{U} is orthogonal by definition.

Note: This is just motivation – we'll cover the math need to understand these topics later!

Why Math for ML?

To best understand A we need B

A	B
Derivation of Principal Component Analysis (PCA)	Linear Algebra <ul style="list-style-type: none">• Vector spaces, Functions and Function Spaces• Matrices and linear operators• Matrix decomposition

Why Math for ML?

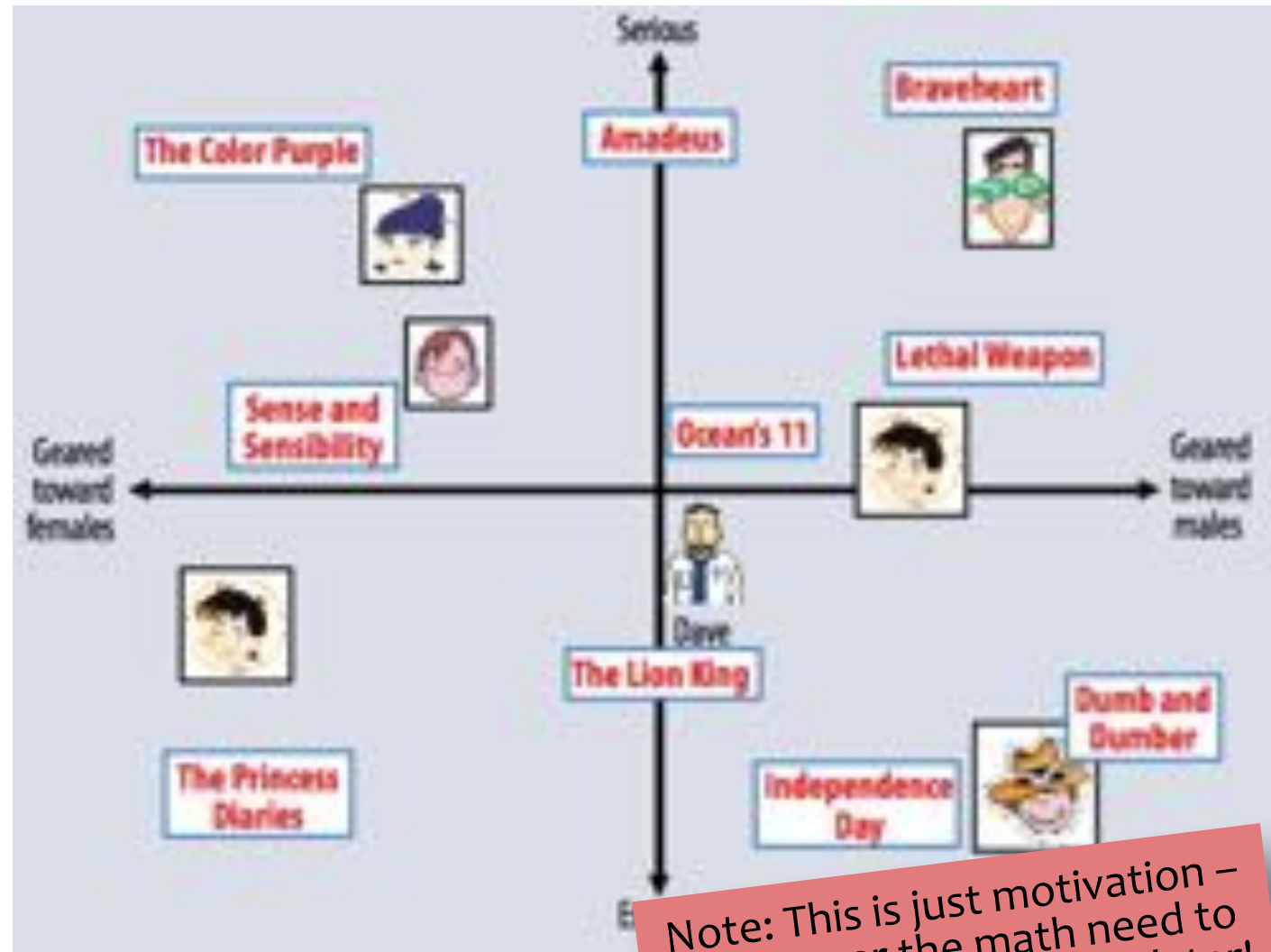
To best understand A we need B

A	B
Derivation of Principal Component Analysis (PCA)	Linear Algebra <ul style="list-style-type: none">• Vector spaces, Functions and Function Spaces• Matrices and linear operators• Matrix decomposition
Gradient-based Matrix Factorization and Collaborative Filtering	Calculus <ul style="list-style-type: none">• Chain-rule; Partial Derivatives;• Matrix Differentials; Second and Higher Differentials

Collaborative Filtering

Latent Factor Methods

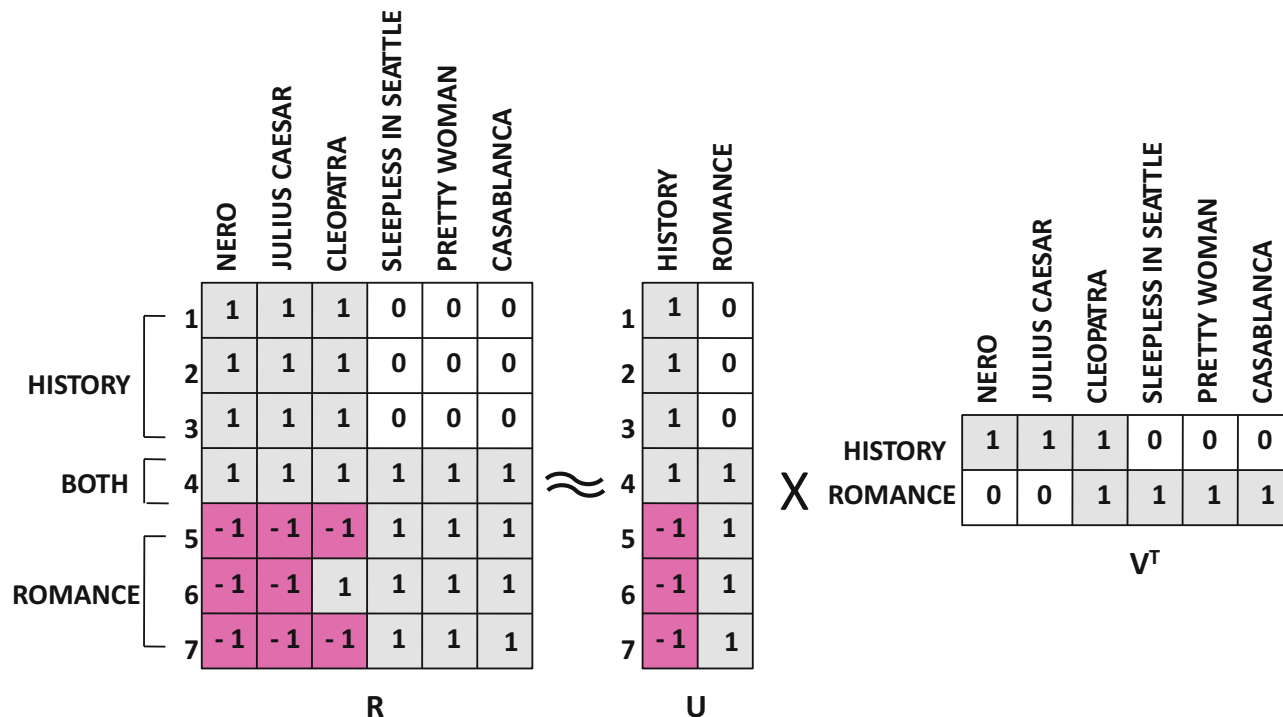
- Assume that both movies and users live in some **low-dimensional space** describing their properties
- Recommend** a movie based on its **proximity** to the user in the latent space



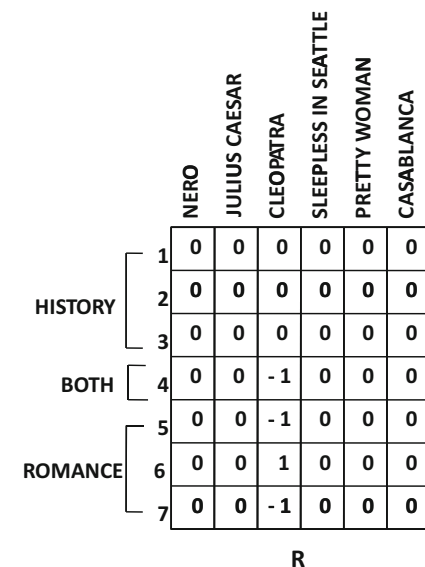
Note: This is just motivation – we'll cover the math need to understand these topics later!

Example: Matrix Factorization

for the Netflix Problem



(a) Example of rank-2 matrix factorization



(b) Residual matrix

Note: This is just motivation – we'll cover the math need to understand these topics later!

Matrix Factorization (with matrices)

- User vectors:

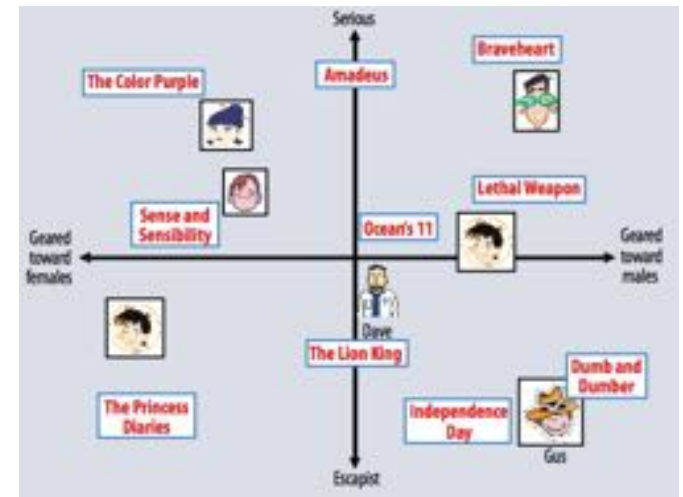
$$(W_{u*})^T \in \mathbb{R}^r$$

- Item vectors:

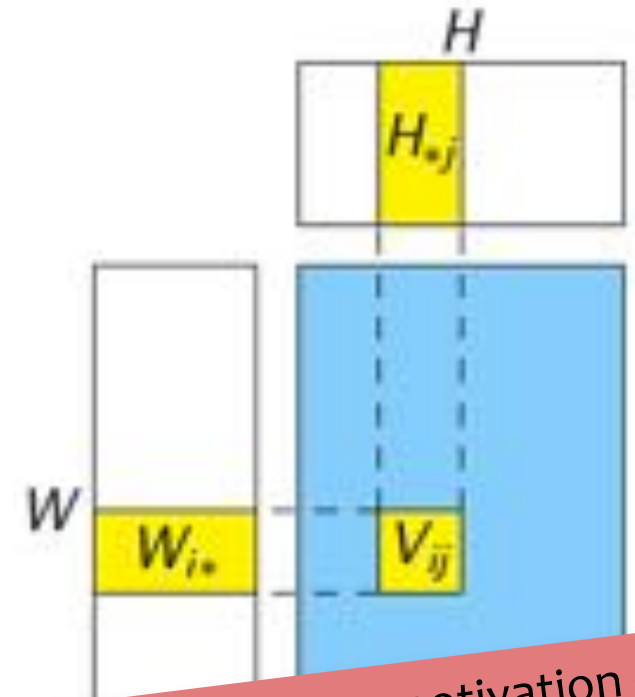
$$H_{*i} \in \mathbb{R}^r$$

- Rating prediction:

$$\begin{aligned} V_{ui} &= W_{u*} H_{*i} \\ &= [WH]_{ui} \end{aligned}$$



Figures from Koren et al. (2009)



Note: This is just motivation – we'll cover the math need to understand these topics later!

Matrix Factorization

(with matrices)

- Stochastic Gradient Descent

require that the loss can be written as

$$L = \sum_{(i,j) \in Z} l(V_{ij}, W_{i*}, H_{*j})$$

Algorithm 1 SGD for Matrix Factorization

Require: A training set Z , initial values W_0 and H_0

while not converged **do** {step}

 Select a training point $(i, j) \in Z$ uniformly at random.

$$W'_{i*} \leftarrow W_{i*} - \epsilon_n N \frac{\partial}{\partial W_{i*}} l(V_{ij}, W_{i*}, H_{*j})$$

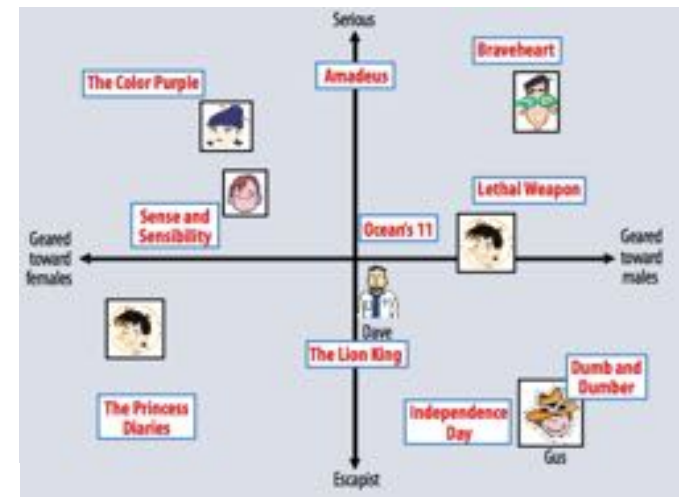
$$H_{*j} \leftarrow H_{*j} - \epsilon_n N \frac{\partial}{\partial H_{*j}} l(V_{ij}, W_{i*}, H_{*j})$$

$$W_{i*} \leftarrow W'_{i*}$$

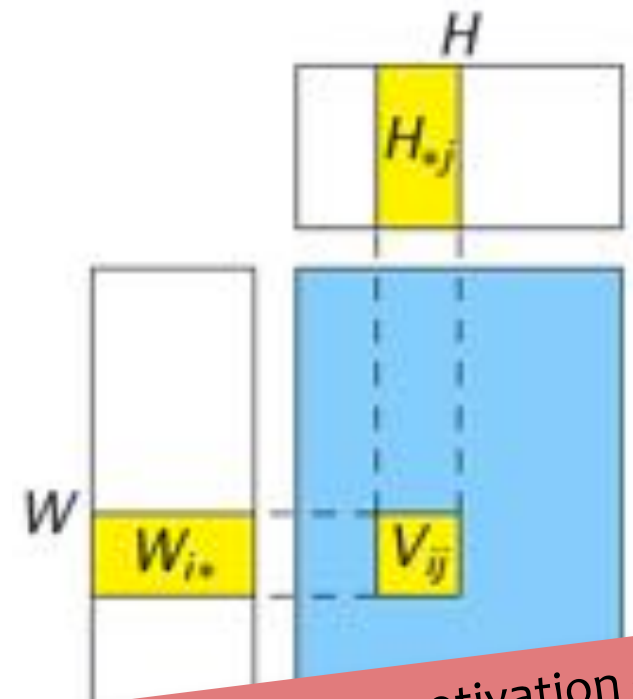
end while

step size

Figure from Gemulla et al. (2011)



Figures from Koren et al. (2009)



Note: This is just motivation – we'll cover the math need to understand these topics later!

Why Math for ML?

To best understand A we need B

A	B
Derivation of Principal Component Analysis (PCA)	Linear Algebra <ul style="list-style-type: none">• Vector spaces, Functions and Function Spaces• Matrices and linear operators• Matrix decomposition
Gradient-based Matrix Factorization and Collaborative Filtering	Calculus <ul style="list-style-type: none">• Chain-rule; Partial Derivatives;• Matrix Differentials; Second and Higher Differentials

Why Math for ML?

To best understand A we need B

A	B
Derivation of Principal Component Analysis (PCA)	Linear Algebra <ul style="list-style-type: none">• Vector spaces, Functions and Function Spaces• Matrices and linear operators• Matrix decomposition
Gradient-based Matrix Factorization and Collaborative Filtering	Calculus <ul style="list-style-type: none">• Chain-rule; Partial Derivatives;• Matrix Differentials; Second and Higher Differentials
Probabilistic Study of Ordinal Regression	Probability <ul style="list-style-type: none">• Events• Discrete/Continuous Random Variables• Mean & Variance; Factorization• Multivariate Distributions

Sentiment Analysis

- **Task:** Given a restaurant description, predict **how many stars** the author would give for the “Overall” rating

Aspects	Ratings	Reviews
Atmosphere	★★☆☆	Heavy, uninspired food, eaten under appall of cigarette smoke. Very slow service, though not unfriendly. There are many better restaurants in Ashland. Not recommended.
Food	★★☆☆	
Value	★★★☆☆	
Service	★★☆☆	
Overall	★★☆☆	
Atmosphere	★★★★☆	I'll have to disagree with Ms. Kitago's take on at least one part of the evening. I believe the chicken Tikka Marsala was slightly dry. Decent portion, but not succulent as I am accustomed to. In addition, the Gulub Jaman is served cold, anathema to this diner. I will agree with Ms. K that the mango lassi was delicious, but overall I believe the price was slightly inflated
Food	★★★★☆	
Value	★★★★☆	
Service	★★★★☆	
Overall	★★★★☆	

Note: This is just motivation – we'll cover the math need to understand these topics later!

Ordinal Regression

NOTEWORTHY STATISTICS (NS)

Ordinal logistic regression

Pamela Warner

Box 1: Glossary of statistical terms used in this article

Binary variable	This is a categorical variable with only two possible values (e.g. case or non-case). Also termed dichotomous or binomial .
Categorical variable	Such a variable has a limited number of distinct values, which might be non-quantitatively descriptive (e.g. hair colour).
Cumulative distribution	For each value of an ordinal or continuous variable, the cumulative frequency presents the accumulated number of occurrences for that or any 'lower' value (rather than indicating simply the frequency of occurrence of that value alone, as in an ordinary frequency distribution). The cumulative frequency for the last (highest) value must therefore encompass the entire sample. Cumulative frequency is often reported not as counts but as a percentage of the total sample, in which case the last value must have a cumulative frequency of 100%.
Degrees of freedom	This can be thought of as the modelling capacity (or independent elements of information) in the dataset.
Logistic regression	This is a method for analysis of the occurrence or not of a particular response value, in relation to potential explanatory variables. What is modelled for each combination of explanatory variables is the logarithm of the odds of that response value (which is termed a logistic transformation). Each association in the model is summarised/estimated in terms of an odds ratio (OR).
Multinomial variable	This is a categorical variable with more than two possible values. Also termed polychotomous .
Odds (of a specified response)	This is the number of occurrences of that response value divided by the number without that value (e.g. cases divided by non-cases).
Odds ratio (OR) for a specified response	For a binary explanatory variable, the OR is calculated as the odds of the specified response in those with the explanatory feature, divided by the odds of that response in those without the explanatory feature. If there is truly no association then the two odds should be approximately equal and the OR approximately 1, which is therefore the value for a 'null association'.
Ordinal variable	This is special semi-quantitative type of categorical variable where the values are conceptually ordered, such as degree of pain (e.g. none, mild, moderate, severe) or effectiveness of contraceptive method used (e.g. none, moderate, high).
Parameter	This is a component in the model, which needs to be estimated from the data, and doing so uses up available degrees of freedom. [The number of parameters needed for a multinomial regression model is a multiple of the number needed for a binary logistic regression model.]
Power	This term is used here, loosely, as the probability of detecting from the study data what is in fact the real situation.
Reference category	The category within a categorical explanatory variable that is chosen as the comparator for calculation of ORs (i.e. denominator odds).
Reference value	This is the value within a <i>response</i> variable that is used as a comparator response value. In multinomial regression because in binary logistic regression there are only two possible response values, the reference value can be assumed to be the only other possible response value.
Response variable	The outcome or dependent variable that is to be modelled/tested.

Note: This is just motivation – we'll cover the math need to understand these topics later!

Ordinal Regression

Ordinal logistic regression

If the response variable Y is ordinal, the categories can be ordered in a natural way such as 'health status good/moderate/bad'. The polytomous logistic regression model can be applied but does not make use of the information about the ordering. One way to take account of the ordering is the use of *cumulative probabilities*, *cumulative odds* and *cumulative logits*. Considering $k+1$ ordered categories, these quantities are defined by

$$P(Y \leq i) = p_1 + \dots + p_i$$

$$\text{odds}(Y \leq i) = \frac{P(Y \leq i)}{1 - P(Y \leq i)} = \frac{p_1 + \dots + p_i}{p_{i+1} + \dots + p_{k+1}}$$

$$\text{logit}(Y \leq i) = \ln \left(\frac{P(Y \leq i)}{1 - P(Y \leq i)} \right), \quad i=1, \dots, k$$

The cumulative logistic model for ordinal response data is given by

$$\text{logit}(Y \leq i) = \alpha_i + \beta_1 X_1 + \dots + \beta_m X_m, \quad i=1, \dots, k$$

Like the polytomous logistic regression model, we have k model equations and one logistic coefficient β_{ij} for each category/covariate combination. Hence, the general cumulative logistic regression model contains a large number of parameters. However, in some cases a more parsimonious model is possible. If the logistic coefficients do not depend on i , we have only one common parameter β_j for each covariate. It follows that the *cumulative odds* are given by

$$\text{odds}(Y \leq i) = \exp(\alpha_i) \exp(\beta_1 X_1 + \dots + \beta_m X_m), \quad i=1, \dots, k$$

which means that the k odds for each cut-off category i differ only with regard to the intercepts α_i ; in other words, the odds are proportional. Hence, McCullagh⁴ used the term *proportional odds model*. The relatively stringent proportional odds assumption may be especially valid in cases where the ordinal response Y is related to an underlying latent continuous variable⁴,

Polytomous logistic regression

If the response variable Y is discrete with more than two categories, for example $Y = \text{marital status}$ defined in the 3 categories 'married', 'divorced, separated or widowed' and 'single', then the standard binary logistic regression model is not applicable. One possible way to handle such situations is to split the categorical response Y in several ways, for example $Y_1 = \text{'married yes/no'}$, $Y_2 = \text{'single yes/no'}$, and to apply binary logistic regression to each dichotomous variable. However, this will result in several different analyses for only one categorical response. A more structured approach is to formulate one model for the categorical response by means of so-called *generalised logits*. Suppose that Y has $k+1$ categories and the probability for category i is given by $P(Y=i) = p_i$ for $i=1, \dots, k+1$. Then the k *generalised logits* are defined by

$$\text{logit}(Y=i) = \ln \left(\frac{p_i}{1 - (p_1 + \dots + p_k)} \right) = \ln \left(\frac{p_i}{p_{k+1}} \right), \quad i=1, \dots, k$$

This means that the *generalised logits* relate the probabilities p_i for the categories $i=1, \dots, k$ to the reference category $k+1$.

For m covariates the general polytomous logistic regression model becomes

$$\text{logit}(Y=i) = \alpha_i + \beta_{1i} X_1 + \dots + \beta_{mi} X_m, \quad i=1, \dots, k$$

Note that the polytomous logistic model is given by k equations if Y has $k+1$ categories and that we have one logistic coefficient β_{ij} for each category/covariate combination. Hence, it is not possible to summarise the effect of a covariate on the response Y by a single measure such as one odds ratio. Although the polytomous model offers the advantage of simultaneously testing the effect of m covariates on $k+1$ categories, polytomous logistic regression involves a cumbersome amount of calculations. It is difficult for physicians to interpret the results of this class of models (DeMaris¹⁸).

Note: This is just motivation – we'll cover the math need to understand these topics later!

Why Math for ML?

To best understand A we need B

A	B
Derivation of Principal Component Analysis (PCA)	Linear Algebra <ul style="list-style-type: none"> • Vector spaces, Functions and Function Spaces • Matrices and linear operators • Matrix decomposition
Gradient-based Matrix Factorization and Collaborative Filtering	Calculus <ul style="list-style-type: none"> • Chain-rule; Partial Derivatives; • Matrix Differentials; Second and Higher Differentials
Probabilistic Study of Ordinal Regression	Probability <ul style="list-style-type: none"> • Events • Discrete/Continuous Random Variables • Mean & Variance; Factorization • Multivariate Distributions

The core content for this course is the **mathematics** (Column B), but you will apply what you learn to **real problems in machine learning** (Column A)

Why Computer Science for ML?

To best understand A we need B

A

B

Why Computer Science for ML?

10-607

To best understand A we need B

A	B
Analysis of Exact Inference in Graphical Models	Computation <ul style="list-style-type: none">• Computational Complexity• Recursion; Dynamic Programming• Data Structures for ML Algorithms

Why Computer Science for ML?

10-607

To best understand A we need B

A	B
Analysis of Exact Inference in Graphical Models	Computation <ul style="list-style-type: none">• Computational Complexity• Recursion; Dynamic Programming• Data Structures for ML Algorithms
Implementation Design of a Deep Learning Library	Programming & Efficiency <ul style="list-style-type: none">• Debugging for Machine Learning• Efficient Implementation / Profiling ML Algorithms

Why Computer Science for ML?

10-607

To best understand A we need B

A	B
Analysis of Exact Inference in Graphical Models	Computation <ul style="list-style-type: none">• Computational Complexity• Recursion; Dynamic Programming• Data Structures for ML Algorithms
Implementation Design of a Deep Learning Library	Programming & Efficiency <ul style="list-style-type: none">• Debugging for Machine Learning• Efficient Implementation / Profiling ML Algorithms
Optimization for Support Vector Machines (SVMs)	Optimization <ul style="list-style-type: none">• Unconstrained Optimization• Preconditioning• Constrained Optimization

The core content for this course is the **computer science** (Column B), but you will apply what you learn to **real problems in machine learning** (Column A)

SYLLABUS HIGHLIGHTS

Syllabus Highlights

The syllabus is located on the course webpage:

<http://www.cs.cmu.edu/~mgormley/courses/606-607-f18>

The **course policies** are **required** reading.

606/607 Syllabus Highlights

- **Grading:** 55% homework, 10% in-class quizzes, 30% final exam, 5% participation
- **Final Exam:**
 - 606: Mini-I final exam week, date TBD
 - 607: Mini-II final exam week, date TBD
- **In-Class Quizzes:** always announced ahead of time
- **Homework:** 4 assignments with written / programming portions
 - 2 grace days for the unexpected
 - Late submissions: 80% day 1, 60% day 2, 40% day 3, 20% day 4
 - No submissions accepted after 4 days w/o extension
 - Extension requests: see syllabus
- **Recitations:** Fridays, same time/place as lecture (optional, interactive sessions)
- **Readings:** required, online, recommended for after lecture
- **Technologies:** Piazza (discussion), Gradescope (homework), Canvas (gradebook only)
- **Academic Integrity:**
 - Collaboration encouraged, but must be documented
 - Solutions must always be written independently
 - No re-use of found code / past assignments
 - Severe penalties (i.e. failure)
- **Office Hours:** posted on Google Calendar on “People” page

606/607 Syllabus Highlights

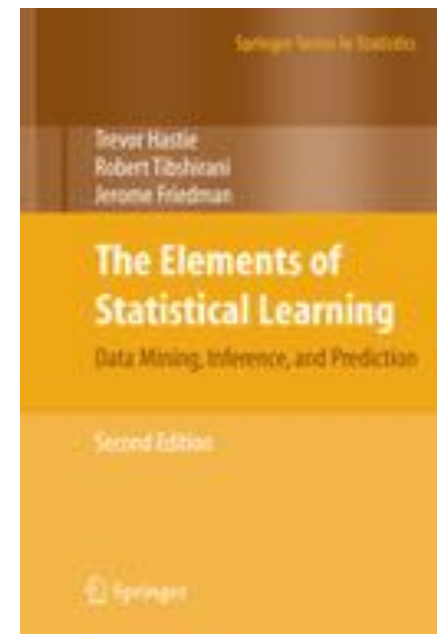
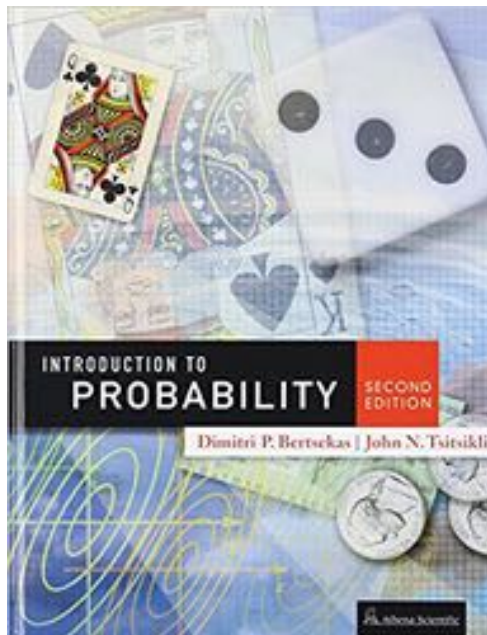
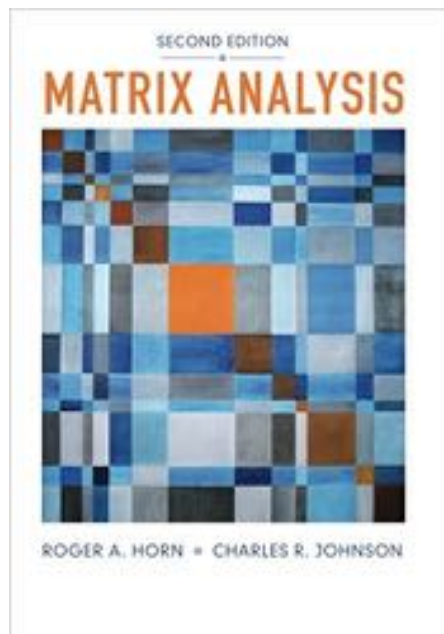
- **Grading:** 55% homework, 10% in-class quizzes, 30% final exam, 5% participation
- **Final Exam:**
 - 606: Mini-I final exam week, date TBD
 - 607: Mini-II final exam week, date TBD
- **In-Class Quizzes:** always announced ahead of time
- **Homework:** 4 assignments with written / programming portions
 - 2 grace days for the unexpected
 - Late submissions: 80% day 1, 60% day 2, 40% day 3, 20% day 4
 - No submissions accepted after 4 days w/o extension
 - Extension requests: see syllabus
- **Recitations:** Fridays, same time/place as lecture (optional, interactive sessions)
- **Readings:** required, online, recommended for after lecture
- **Technologies:** Piazza (discussion), Gradescope (homework), Canvas (gradebook only)
- **Academic Integrity:**
 - Collaboration encouraged, but must be documented
 - Solutions must always be written independently
 - No re-use of found code / past assignments
 - Severe penalties (i.e. failure)
- **Office Hours:** posted on Google Calendar on “People” page

Lectures

- You should ask lots of questions
 - Interrupting (by raising a hand) to ask your question is strongly encouraged
 - Asking questions later on Piazza is also great
- When I ask a question...
 - I want you to answer
 - Even if you don't answer, think it through as though I'm about to call on you
- Interaction improves learning (both in-class and at my office hours)

Textbooks

These are optional, but highly recommended as an alternate presentation of the material



EXPECTED BACKGROUND

Expected Background

10-606 (Math Background 4 ML)

You should be familiar with some of the following...

- Calculus:
 - can take scalar derivatives
 - can solve scalar integrals
- Linear Algebra:
 - know basic vector operations
 - seen matrix multiplication
- Probability:
 - seen the basics: conditioning, Bayes Rule, etc.
- Programming:
 - know some Python**OR**
 - have sufficient programming background to pick up the basics of Python

But we'll offer practice to make sure you can catch up on your weaker areas

10-607 (CS Background 4 ML)

You should...

- be comfortable with all the topics listed for 10-606
- ideally, have the mathematical maturity of someone who completed 10-606 **because it will aide in understanding the motivating examples from machine learning**

That said, the content of 10-607 is designed stand alone

PRELIMINARIES

Sets

Chalkboard

- Definitions: Set, element of, equality, subset
- Example: Sets of sets
- Set builder notation
- Python list/set comprehensions
- Exercise: Set builder notation
- Definitions: Union, intersection, difference, complement
- Exercise: Set complement
- Tuples and set product
- Exercise: Set product

Data Types and Functions

Chalkboard

- Data types, structs, unions
- Tagged unions
- Exercise: Tagged unions
- Functions
- Anonymous functions
- Exercises: Functions