CPSC 540: Machine Learning

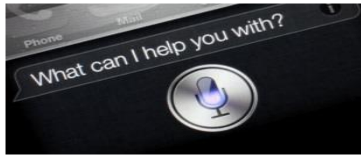
Mark Schmidt

University of British Columbia, Winter 2016

www.cs.ubc.ca/~schmidtm/Courses/540-W16

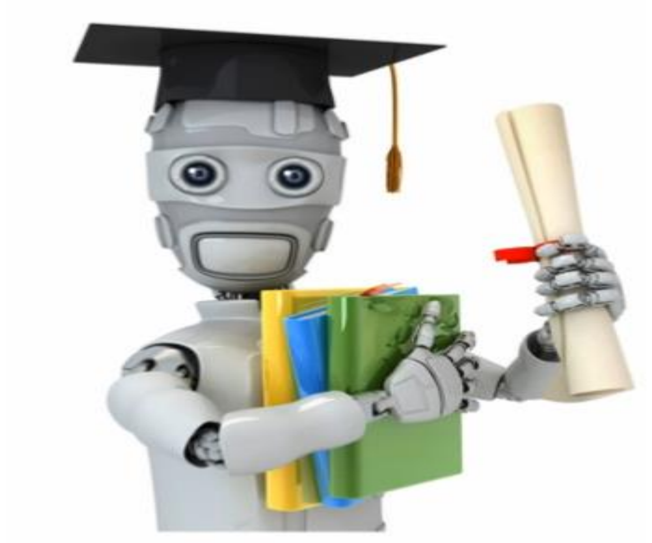
Big Data Phenomenon

- We are collecting and storing data at an unprecedented rate.
- Examples:
  - News articles and blog posts.
  - YouTube, Facebook, and WWW.
  - Credit cards transactions and Amazon purchases.
  - Gene expression data and protein interaction assays.
  - Maps and satellite data.
  - Large hadron collider and surveying the sky.
  - Phone call records and speech recognition results.
  - Video game worlds and user actions.



Machine Learning

- What do you do with all this data?
  - Too much data to search through it manually.
- But there is valuable information in the data.
  - Can we use it for fun, profit, and/or the greater good?
- Machine learning: use computers to automatically detect patterns in data and make predictions or decisions.
- Most useful when:
  - Don't have a human expert.
  - Humans can't explain patterns.
  - Problem is too complicated.

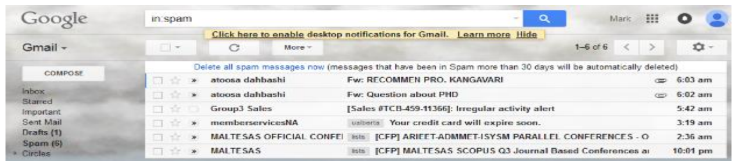


Machine Learning vs. Statistics

- Machine learning (ML) is very similar to statistics.
  - A lot of topics overlap.
- But ML places more emphasis on:
  1. Computation and large datasets.
  2. Predictions rather than descriptions.
  3. Non-asymptotic performance.
  4. Models that work across domains.
- The field is growing very fast:
  - ~2500 attendees at NIPS 2014, ~4000 at NIPS 2015.
  - Influence of $$$, too.

Applications

- Spam filtering.
- Credit card fraud detection.
- Product recommendation.
- Motion capture.
- Machine translation.
- Speech recognition.
- Face detection.
- Object detection.
- Sports analytics.

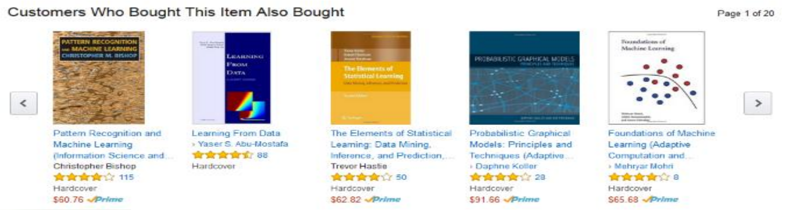


Click here to enable desktop notifications for Gmail. Learn more Hide

Delete all spam messages now (messages that have been in Spam more than 30 days will be automatically deleted)

atoosa dahbashi	Fw: RECOMMEN PRO. KANGAVARI	6:03 am
atoosa dahbashi	Fw: Question about PHD	6:02 am
Group3 Sales	[Sales #TCB-459-11366]: Irregular activity alert	5:42 am
memberservicesNA	ualberta Your credit card will expire soon.	3:19 am
MALTESAS OFFICIAL CONFE	lists [CFP] ARIEET-ADMMET-ISYSM PARALLEL CONFERENCES - O	2:36 am
MALTESAS	lists [CFP] MALTESAS SCOPUS Q3 Journal Based Conferences ai	10:01 pm

Transaction Date	Posted Date	Transaction Details	Debit	Credit
Aug. 27, 2015	Aug. 28, 2015	BEAN AROUND THE WORLD VANCOUVER, BC	$10.95	



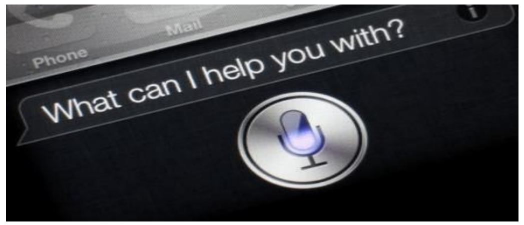
Customers Who Bought This Item Also Bought

Page 1 of 20

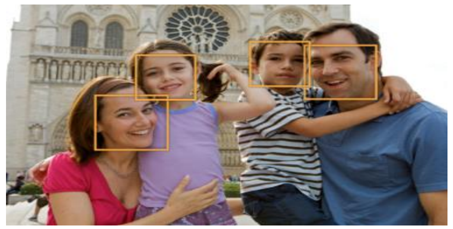
Pattern Recognition and Machine Learning (Information Science and…	Learning From Data › Yaser S. Abu-Mostafa	The Elements of Statistical Learning: Data Mining, Inference, and Prediction,…	Probabilistic Graphical Models: Principles and Techniques (Adaptive…	Foundations of Machine Learning (Adaptive Computation and…
Christopher Bishop	88	Trevor Hastie	› Daphne Koller	› Mehryar Mohri
115	Hardcover	50	28	8
Hardcover		Hardcover	Hardcover	Hardcover
$60.76 Prime		$62.82 Prime	$91.66 Prime	$65.68 Prime

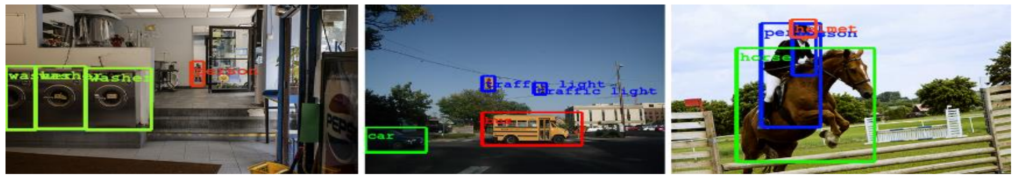
English – detected	French
Data mining.	L'exploration de données .

Open in Google Translate



What can I help you with?

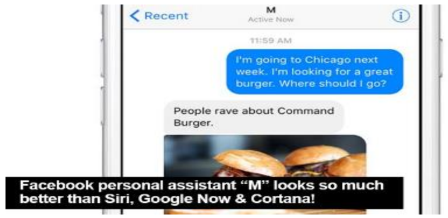


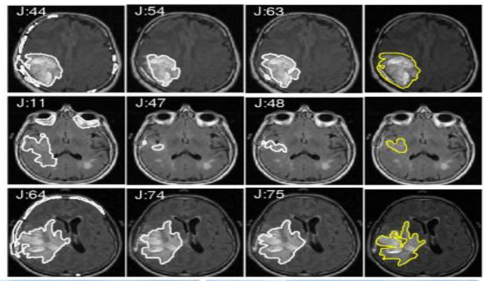


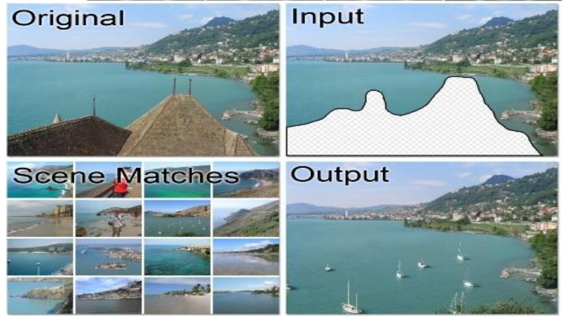
KLAY THOMPSON
VERY GOOD FROM HERE!
MEH...
BY: @KIRKGOLDSBERRY
GRANTLAND

Applications

- Personal Assistants.
- Medical imaging.
- Self-driving cars.
- Scene completion.
- Image search and annotation.
- Artistic rendering.
- Your research?


Recent
M
Active Now
11:59 AM
I'm going to Chicago next week. I'm looking for a great burger. Where should I go?
People rave about Command Burger.
Facebook personal assistant "M" looks so much better than Siri, Google Now & Cortana!

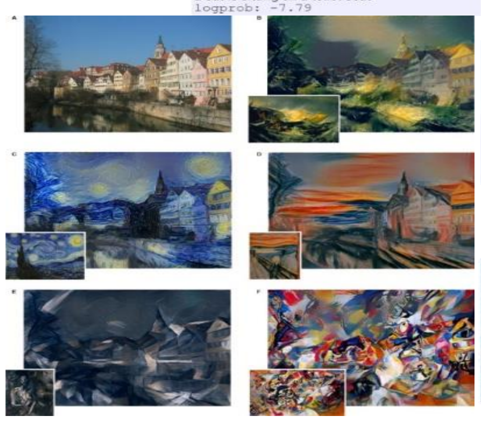




a cat is sitting on a toilet seat
logprob: -7.79

a display case filled with lots of different types of donuts
logprob: -7.78

a group of people sitting at a table with wine glasses
logprob: -6.71



Horizon
Trees
Leaves
Towers & Pagodas
Buildings
Birds & Insects

CPSC 340 and CPSC 540

- There are two ML classes: CPSC 340 and 540.
  - Roughly structured as one full-year course.
- CPSC 340:
  - Introductory course on data mining and ML.
  - Emphasis on applications of ML.
  - Useful techniques you can apply to your research.
  - I strongly recommend taking CPSC 340 first.
- CPSC 540:
  - Research-level ML methods and theory.
  - Not an introductory course:
    - Assumes familiarity with basic ML concepts.
    - Stronger math/CS background
    - Much more work.
    - Many standard topics are not covered:
      - Random forests, clustering, collaborative filtering, data visualization, etc.

Course Outline

- 2-4 lectures on each of the following:
  - Linear models (also covered in CPSC 340).
  - Advanced linear models.
  - Density estimation.
  - Graphical models.
  - Bayesian methods.
  - Deep learning (also covered in CPSC 340).
  - Causal, active, and online learning.
  - Reinforcement learning.
  - Learning theory.
- For an overview of topics covered in 340 and 540 see here:
  - http://www.cs.ubc.ca/~schmidtm/Courses/340-F15/L35.pdf

Math Prerequisites

- Research-level ML involves a lot of math.
- You should be comfortable with:
  - Linear algebra: vectors, matrices, eigenvalues.
  - **Probability**: conditional probability, expectations.
  - **Multivariate calculus**: gradients, optima.
  - Proof strategies and filling in derivation details.
- Suggested courses: Math 200, 220, 221, 302.

- "I didn't really feel prepared for this course. I had never really done vector calculus before."

Computer Science Prerequisites

- ML places a big emphasis on computation.
- You should be comfortable with:
  - Data structures: pointers, trees, heaps, hash maps, graphs.
  - **Algorithms and complexity**:
    - Big-O, divide + conquer, randomized algorithms, dynamic programming, proving NP-hard.
  - Scientific computing: matrix factorization, conditioning.
- Suggested courses: CPSC 221, 302, 320:
  - "I have programming experience in my work/research/courses" is not enough.

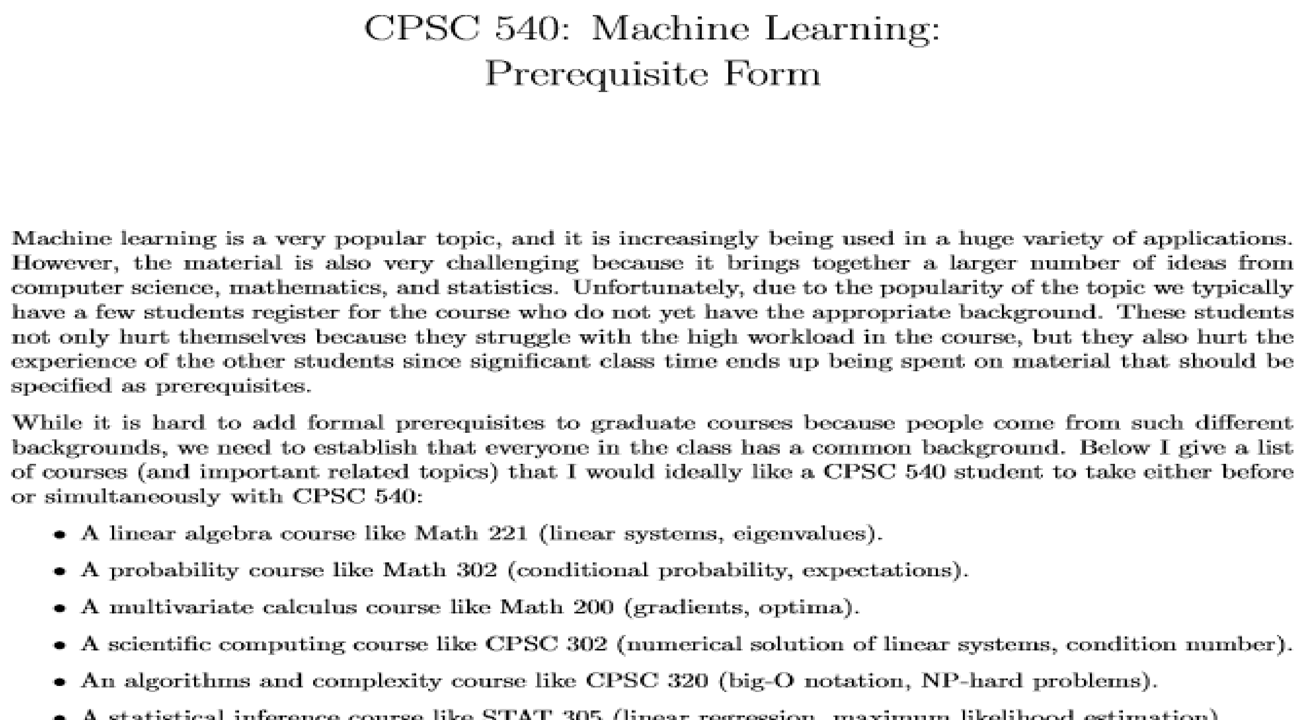
- "It is taught in a manner very hard and intimidating for those who are not in computer science."

Stat/ML Prerequisites

- This is not an introductory ML course.
- You should be comfortable with:
  - Linear regression.
  - Cross-validation.
  - Regularization.
  - Logistic regression.
- We will only cover these topics briefly and then build on them.
  - If you don’t know these, you will fall behind quickly.
- Suggested courses: Stat 306 or CPSC 340.
  - “I did Coursera” is not enough.

Prerequisite Form

- All students must submit the prerequisite form.
  - CS and ECE grad students: submit in class by January 14.
  - All others: submit to enroll in course.



CPSC 540: Machine Learning:
Prerequisite Form

Machine learning is a very popular topic, and it is increasingly being used in a huge variety of applications. However, the material is also very challenging because it brings together a larger number of ideas from computer science, mathematics, and statistics. Unfortunately, due to the popularity of the topic we typically have a few students register for the course who do not yet have the appropriate background. These students not only hurt themselves because they struggle with the high workload in the course, but they also hurt the experience of the other students since significant class time ends up being spent on material that should be specified as prerequisites.

While it is hard to add formal prerequisites to graduate courses because people come from such different backgrounds, we need to establish that everyone in the class has a common background. Below I give a list of courses (and important related topics) that I would ideally like a CPSC 540 student to take either before or simultaneously with CPSC 540:

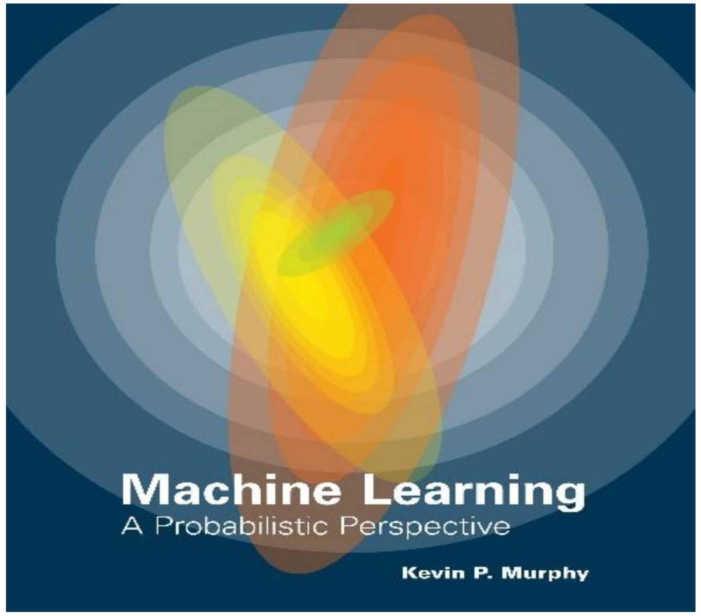
- A linear algebra course like Math 221 (linear systems, eigenvalues).
- A probability course like Math 302 (conditional probability, expectations).
- A multivariate calculus course like Math 200 (gradients, optima).
- A scientific computing course like CPSC 302 (numerical solution of linear systems, condition number).
- An algorithms and complexity course like CPSC 320 (big-O notation, NP-hard problems).
- A statistical inference course like STAT 305 (linear regression, maximum likelihood estimation).

Auditing and Recording

- Auditing 540, an excellent option:
  - Pass/fail on transcript rather than grade.
  - Do 1 assignment or write a 2-page report on one technique from class or attend > 90% of classes.
  - But please do this officially:
    - http://students.ubc.ca/enrolment/courses/academic-planning/audit
- About recording lectures:
  - Do not record without permission.
  - All class material will be available online.

Textbook and Other Optional Reading

- No textbook covers all course topics.
- The closest is Kevin Murphy's "Machine Learning".



- For each lecture:
  - I'll give relevant sections from this book.
  - I'll give other related online material.
- There is a list of related courses on the webpage.

Grading

- Course grades will be split evenly between:
  - 5 assignments (written and Matlab programming).
  - Midterm (currently scheduled for March 17).
  - Course project (currently scheduled to be due April 19).

- Your department should have Matlab access:
  - If not, you can get a CS guest account or buy it through the bookstore.

Assignments

- Due at the start of Tuesday classes:
  - A1: January 19 (2 weeks), February 2, February 23, March 8, March 29.
- Start early, the assignments are a lot of work:
  - “This course's workload was a bit more than I would have liked. It seems like this course takes twice the amount of time as another course.”
- You can do assignments in groups of 1 to 3.
  - Hand in one assignment for the group.
  - But each member should still know the material.

Late Assignment Policy

- You can have up to 3 total 'late classes':
  - Handing in on Thursday counts as 1.
  - Handing in on next Tuesday counts as 2.
  - Examples:
    - you can hand in A1, A4, and A5 one day late.
    - you can hand in A2 two days late and A4 one day late.
    - you can hand in A1 three days late, and all others on time.
      - If you need multiple late days for A1, consider dropping course.
  - Due to midterm, you cannot use multiple late days on A4.
  - Beyond 3 late classes, you will get a mark of 0.

Getting Help

- Piazza for assignment/course questions:
  - piazza.com/ubc.ca/winterterm12015/cpsc340
- Office hours:
  - Wednesdays 11:30-12:30 (ICICS 193), or by appointment.
- Optional weekly tutorials:
  - Run by TAs covering related material.
  - Fridays 3:00-4:00 and 4:00-5:00 (DMP 101).
- Optional help sessions:
  - Only on weeks when assignment are due.
  - Probably Mondays 3:30-5.
- Teaching Assistants:
  - Reza Babanezhad.
  - Alireza Shafaei.
  - Sharan Vaswani.

Midterm and Course Project

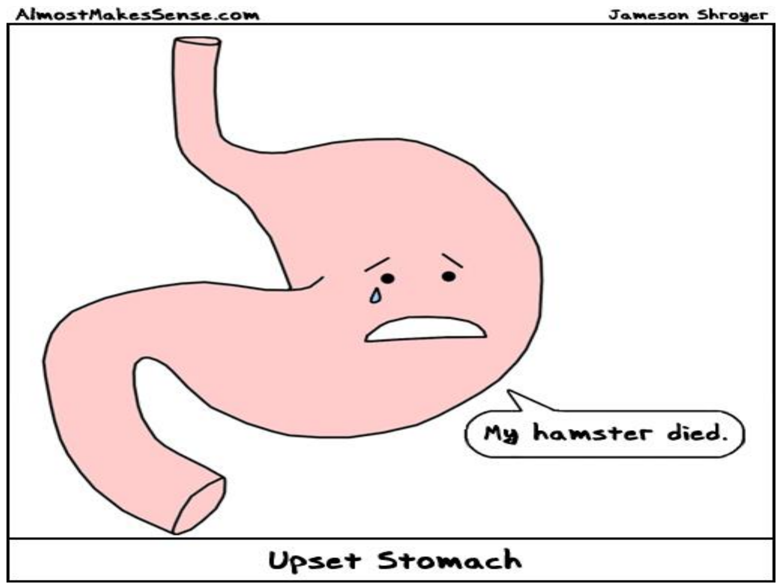
- Midterm details:
  - In class, scheduled for March 17.
  - Closed book, two-page double-sided ‘cheat sheet’.
  - Given a list of things you need to know how to do.
  - Mostly minor variants on assignment questions.
  - No requirement to pass the midterm.
- Let me know ASAP if you must miss the exam.

- Course projects can be done in groups of 2-3:
  - A lot of flexibility in possible topics:
    - Literature review, coding, experiments, application, theory.
  - More details coming later in term.

(pause)

Motivating Example: Food Allergies

- You frequently start getting an upset stomach



- You suspect an adult-onset food allergy.

http://www.cliparthut.com/upset-stomach-clipart-cn48e5.html

Motivating Example: Food Allergies

- You start recording food and IgE levels each day:

	Egg	Milk	Fish	Wheat	Shellfish	Peanuts	…		IgE
Day 1	0	0.7	0	0.3	0	0		⇒	700
Day 2	0.3	0.7	0	0.6	0	0.01		⇒	740
Day 3	0	0	0	0.8	0	0		⇒	50
Day 4	0.3	0.7	1.2	0	0.10	0.01		⇒	950

- We want to write a program that:
  - Takes food levels for the day as an input.
  - Predicts IgE level for the day as the output.
- But foods interact: ‘formula’ mapping foods to IgE is hard to find:
  - Given the data, we could use machine learning to write this program.
  - The program will predict target (IgE levels) given features (food levels).

Supervised Learning

- This is an example of supervised learning:
  - Input is ‘n’ training examples (x_i, y_i).

$$n\left\{\begin{array}{l} x_1 = \overbrace{[\;0 \quad 0.7 \quad 0 \quad 0.3 \quad 0 \quad 0\;]}^{d} \\ x_2 = [\;0.3 \quad 0.7 \quad 0 \quad 0.6 \quad 0 \quad 0.01\;] \\ x_3 = [\;0 \quad 0 \quad 1.2 \quad 0 \quad 0.1 \quad 0.01\;] \\ \qquad \vdots \end{array}\right. \qquad \begin{array}{l} y_1 = 700 \\ y_2 = 740 \\ y_3 = 50 \\ \quad \vdots \end{array}$$

  - x_i is the features for example ‘i’ (we’ll use ‘d’ as the number of features).
    - In this case, the quantities of food eaten on day ‘i’.
  - y_i is target for example ‘i’.
    - In this case, the level of IgE.
  - Output is a function mapping from x_i space to y_i space.

$$f(\text{egg}, \text{milk}, \text{fish}, \text{wheat}, \text{shellfish}, \text{peanuts}) = \text{IgE}. \qquad f(x_1) = f(0, 0.7, 0, 0.3, 0, 0) = 700 = y_1$$

Supervised Learning

- Supervised learning is most successful ML method:
  - Spam filtering, Microsoft Kinect, speech recognition, object detection, etc.
- Most useful when:
  - You don't know how to map from inputs to outputs.
  - But you have a lot of input-to-output examples.

If $w_6 > 0$

more peanuts $\Rightarrow$ more IgE

- When y_i is continuous, it's called regression.
- Today, we consider the special case of linear regression:

$$\underbrace{\widehat{IgE}}_{\text{predicted value}} = \underbrace{w_1}_{\text{weight for 'eggs'}}\overset{\text{feature}}{(eggs)} + w_2(milk) + \underbrace{w_3}_{\text{weight for 'fish'}}(fish) + w_4(wheat) + w_5(shellfish) + w_6(peanuts)$$

Linear Regression

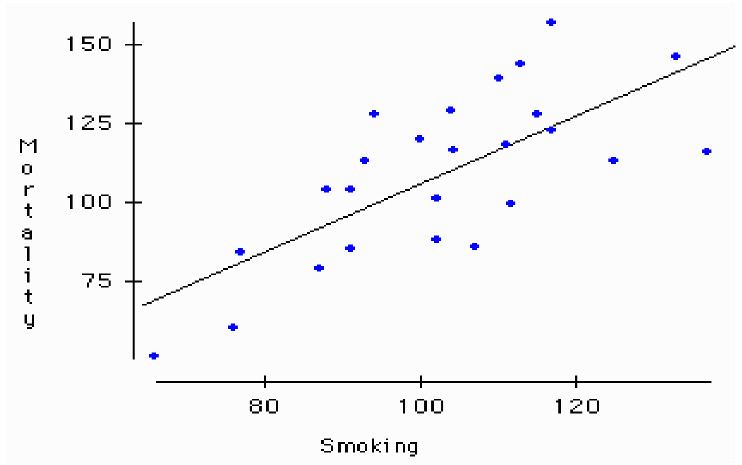
- Linear regression:
  - Prediction is weighted sum of features:

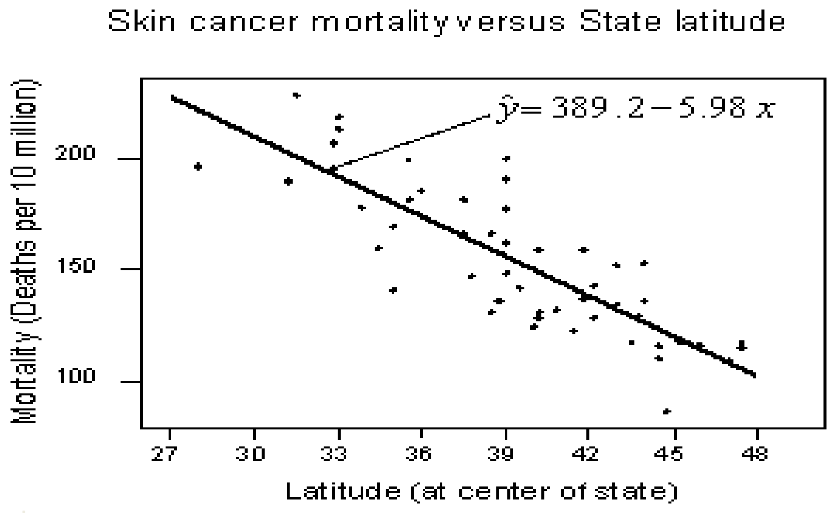
$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3} + \cdots + w_d x_{id}$$

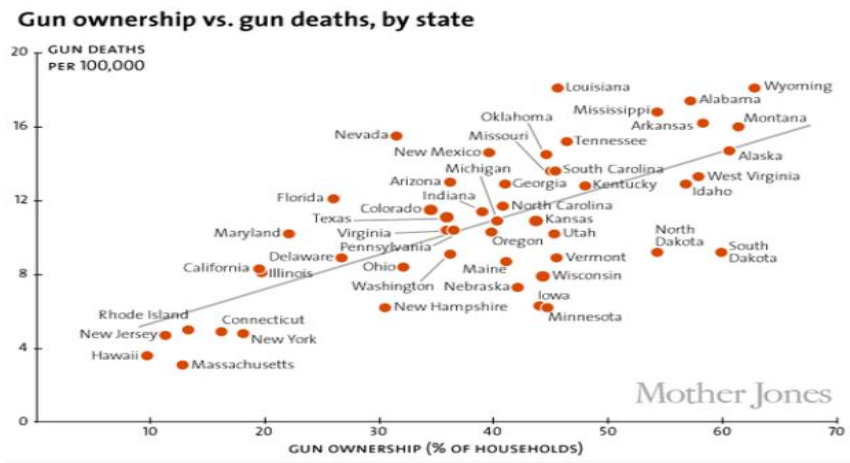
prediction for example 'i'

"weight" of feature 2

feature 2 for example 'i'







http://www.cvgs.k12.va.us:81/digstats/main/inferant/d_regrs.html
https://onlinecourses.science.psu.edu/stat501/node/11
http://www.vox.com/2015/10/3/9444417/gun-violence-united-states-america

Least Squares

- Supervised learning goal:

If we have $x_i = \begin{array}{cccccc} \text{egg} & \text{milk} & \text{fish} & \text{wheat} & \text{shellfish} & \text{peanut} \\ [\,0 & 0.7 & 0 & 0.3 & 0 & 0\,] \end{array}$ and $y_i = 700$,

then we have

$$\hat{y}_i = w_1(0) + w_2(0.7) + w_3(0) + w_4(0.3) + w_5(0) + w_6(0),$$

and we <u>want</u> $(\hat{y}_i - 700)$ to be <u>small</u>.

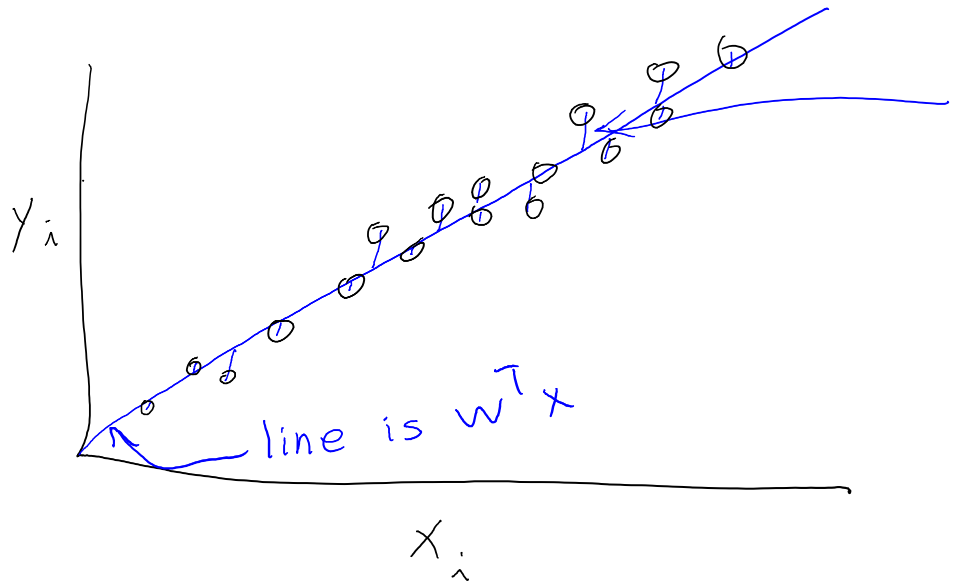
- Can we choose weights to make this happen?
- The classic way to do is minimize square error:

Find $\hat{w}$ that minimizes $(\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + (\hat{y}_3 - y_3)^2 + \cdots$

This should make the average $(\hat{y}_i - y_i)$ <u>small</u>.

Least Squares Objective

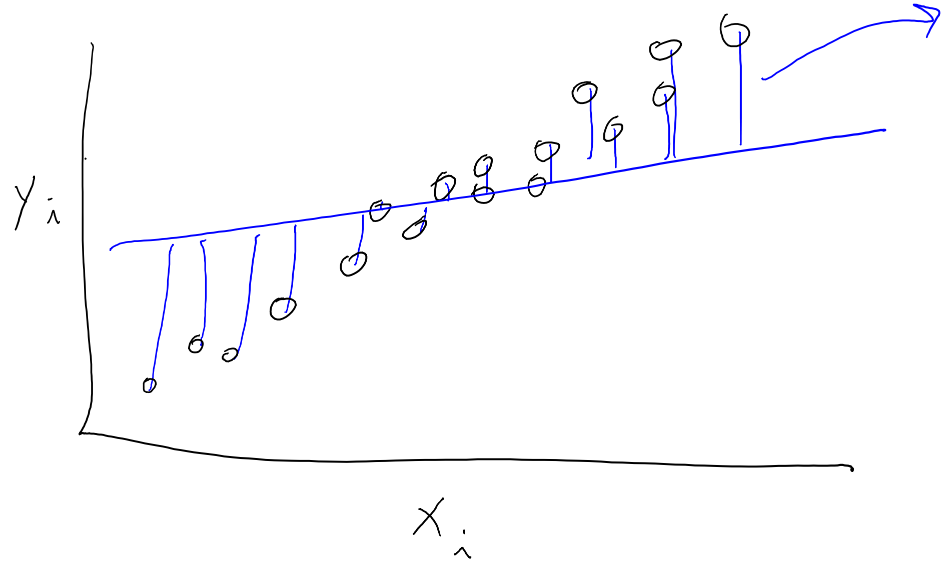
- Classic procedure: minimize squared error:



"errors" $(y_i - \hat{y}_i)$, if these are <u>small</u> then the line predicts y_i accurately.

Least Squares Objective

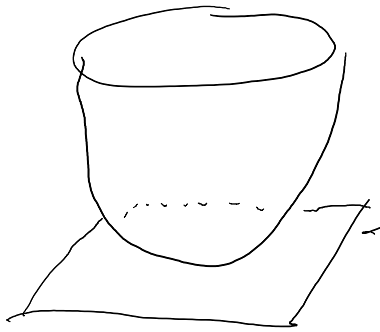
- Classic procedure: minimize squared error:



If these vertical distances, $(y_i - \hat{y}_i)$ are large, then the line does a bad job of predicting y_i.

Least Squares Objective

- Why squared error?
  - There is some theory justifying this choice:
    - Errors follow a normal distribution.
    - Central limit theorem.
  - Computation: quadratic function, so easy to minimize.
- How do we calculate the optimal ‘w’?
  - The error is a convex quadratic function of ‘w’.
  - We can take the gradient and set it to zero.



at minimizer, gradient is (0,0).

Least Squares (Vector Notation)

- So our objective is to minimize

$$(\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + (\hat{y}_3 - y_3)^2 + \cdots + (\hat{y}_n - y_n)^2$$

  in terms of 'w', where we have:

$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3} + \cdots + w_d x_{id}$$

- Written with summation notation:

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \quad \text{with} \quad \hat{y}_i = \sum_{j=1}^{d} w_j x_{ij} = w^T x_i$$

$$w = \underset{d \times 1}{\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_d \end{bmatrix}}$$

- Observe that prediction is inner product.
- This lets us write objective as:

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

Least Squares (Matrix Notation)

- So least squares using vectors ‘w’ and ‘x_i’ is:

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

- To derive solution, need matrix notation:

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \; \| Xw - y \|^2$$

- Where:
  - Each row of feature matrix ‘X’ is an ‘x_i^T’.
  - Element ‘i’ of target vector ‘y’ is ‘y_i’.
  - ||z|| is the Euclidean norm.

$$\begin{array}{cc} & n \times d \\ X = & \begin{bmatrix} \text{—— } x_1^T \text{ ——} \\ \text{—— } x_2^T \text{ ——} \\ \text{—— } x_3^T \text{ ——} \\ \vdots \\ \text{—— } x_n^T \text{ ——} \end{bmatrix} \end{array} \qquad \begin{array}{cc} & n \times 1 \\ y_1 = & \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} \end{array}$$

$$\|z\|^2 = \sum_{j=1}^{d} z_j^2 = \sum_{j=1}^{d} z_j z_j = z^T z \xleftarrow{\text{observe}} \|z\| = \sqrt{\sum_{j=1}^{d} z_j^2}$$

Least Squares (Matrix Notation)

Observe that $w^Tx_1 = x_1^Tw$ (scalar)

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \sum_{i=1}^n (w^Tx_i - y_i)^2$$

$$\Updownarrow$$

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \sum_{i=1}^n r_i^2 \quad \text{where } r_i = w^Tx_i - y_i \text{ "residual"}$$

$$\Updownarrow$$

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \|r\|^2 \Leftrightarrow \boxed{\underset{w \in \mathbb{R}^d}{\text{minimize}} \|Xw - y\|^2}$$

$$r = \begin{bmatrix} w^Tx_1 - y_1 \\ w^Tx_2 - y_2 \\ w^Tx_3 - y_3 \\ \vdots \\ w^Tx_n - y_n \end{bmatrix} = \underbrace{\begin{bmatrix} x_1^Tw \\ x_2^Tw \\ x_3^Tw \\ \vdots \\ x_n^Tw \end{bmatrix}}_{Xw} - \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}}_{y} = Xw - y$$

- Where:
  - Each row of feature matrix 'X' is an 'x_i^T'.
  - Element 'i' of target vector 'y' is 'y_i'.
  - ||z|| is the Euclidean norm.

$$X = \begin{bmatrix} \text{———} & x_1^T & \text{———} \\ \text{———} & x_2^T & \text{———} \\ \text{———} & x_3^T & \text{———} \\ & \vdots & \\ \text{———} & x_n^T & \text{———} \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$$

$$\|z\|^2 = \sum_{j=1}^d z_j^2 = \sum_{j=1}^d z_j z_j = z^Tz \xleftarrow{\text{observe}} \|z\| = \sqrt{\sum_{j=1}^d z_j^2}$$

Gradient Vector

- Deriving least squares solution:
  - Set gradient equal to zero and solve for 'w'.
- Recall gradient is vector of partial derivatives:

$$\nabla f(w) = \begin{bmatrix} \frac{\partial f}{\partial w_1} \\ \frac{\partial f}{\partial w_2} \\ \frac{\partial f}{\partial w_3} \\ \vdots \\ \frac{\partial f}{\partial w_d} \end{bmatrix}$$

- Gradients appear a lot in ML.

Digression: Linear Functions

- A linear function of ‘w’ is a function of the form:

$$f(w) = \underset{\text{vector}}{a}^T w + \underset{\text{scalar}}{\beta}$$

- Gradient of linear function in matrix notation:
  1. Convert to summation notation: $$f(w) = \sum_{i=1}^{d} a_i w_i + \beta$$
  2. Take generic partial derivative: $$\frac{\partial}{\partial w_i} f(w) = a_i$$
  3. Assemble into vector and simplify: $$\nabla f(w) = \begin{bmatrix} \frac{\partial}{\partial w_1} f(w) \\ \frac{\partial}{\partial w_2} f(w) \\ \vdots \\ \frac{\partial}{\partial w_d} f(w) \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_d \end{bmatrix} = a$$

Digression: Quadratic Functions

- A quadratic function 'w' is a function of the form:

$$f(w) = \frac{1}{2}\underbrace{w^T A w}_{A:\ d \times d \text{ matrix}} + \underbrace{b^T}_{\text{vector}} w + \underbrace{\gamma}_{\text{scalar}}$$

Generalizes $\frac{d}{dw}\left[aw^2\right] = 2aw$

- Gradient of quadratic function in matrix notation:

Let $f(w) = w^T A w$.

1. Summation notation:

$$f(w) = w^T \underbrace{\begin{bmatrix} \sum_{j=1}^d a_{1j} w_j \\ \sum_{j=1}^d a_{2j} w_j \\ \vdots \\ \sum_{j=1}^d a_{dj} w_j \end{bmatrix}}_{Aw} = \sum_{i=1}^d \sum_{j=1}^d w_i a_{ij} w_j$$

$$= \sum_{i=1}^d \left(a_{ii} w_i^2 + \sum_{j \neq i} w_i a_{ij} w_j\right)$$

2. Generic partial:

$$\frac{\partial f}{\partial w_i}\left[w^T A w\right] = 2a_{ii} w_i + \sum_{j \neq i} w_j a_{ji} + \sum_{j \neq i} a_{ij} w_j$$

$$= \sum_{j=1}^d w_j a_{ji} + \sum_{j=1}^d a_{ij} w_j$$

3. Assemble/simplify:

$$\nabla f(w) = \begin{bmatrix} \sum_{j=1}^d w_j a_{j1} + \sum_{j=1}^d a_{1j} w_j \\ \sum_{j=1}^d w_j a_{j2} + \sum_{j=1}^d a_{2j} w_j \\ \vdots \\ \sum_{j=1}^d w_j a_{jd} + \sum_{j=1}^d a_{dj} w_j \end{bmatrix}$$

$$= A^T w + Aw$$

$$= (A^T + A)w$$

If symmetric:

$$= (A + A)w$$

$$= 2Aw$$

Summary

- Machine learning: automatically detecting patterns in data to help make predictions and/or decisions.
- CPSC 540: graduate-level 2nd or 3rd course on this topic.
- Supervised learning: using data to learn input:output map.
- Least squares: classic approach to linear regression.
- Quadratics: frequently arise in ML (minimizing is linear system).

- Next time: is this actually "learning" in any meaningful sense?