WEEK 2 SECTION: LOGISTIC REGRESSION

ALEX MEI | SPRING 2023 | NATURAL LANGUAGE PROCESSING

OFFICE HOURS: MONDAY 4 – 5 PM OUTSIDE HENLEY 2113

cs.ucsb.edu/~alexmei/cs190i.html

About Me

Research:

* Responsible Machine Learning: AI Transparency and Safety
* Natural Language Generation: Visual Augmentation

Industry:

* Internships: Procore, Amazon, Benchling, Two Sigma
* Interests: Machine Learning Research, Deep Learning

Etcetera:

* UCSB CS BSMS (2022, 2023), ERSP (2020)
* Hobbies: Cooking, Pokemon TCG, Reality Competitions

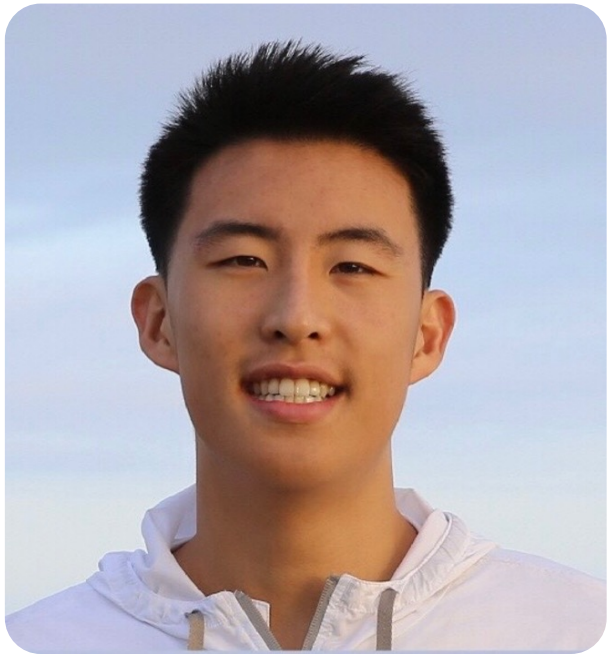
Resources

https://sites.cs.ucsb.edu/~alexmei/cs190i.html

https://piazza.com/class/lfxgp66zcut509

http://william.cs.ucsb.edu/courses/index.php/Spring_2023_CS190I_Introduction_to_Natural_Language_Processing

Tree Mail



* HW1 Due Tuesday April 25th at 1 PM
* Monday April 17th OH rescheduled to 1 - 2 PM
* Don't miss a legendary alumni talk by the 🐐 **Tony Sun**
  * Monday April 17th 3 - 5 PM in Henley 1010 Lecture Hall
  * Talks on (1) college insights and (2) machine learning operations
  * Lead author of "Mitigating Gender Bias in NLP" with over 360+ citations
  * Ranked 1st in class UCSB '21 → Stanford M.S. in Computer Science + Google SWE

Setting: The Restaurant Scoring Problem

* **Input:** features (food taste, service quality, etc.) – $x \in R^d \equiv X$
* **Output:** restaurant rating score – $y \in R^1 \equiv Y$
* **Given:** customers reviews – $D : \{(x_1, y_1), \ldots, (x_n, y_n)\}$
* **Goal:** learn relationship that models output for given input – $f : X \rightarrow Y$

Osha Thai is right by the waterfront in SF, a very lovely location! I ordered their Pad Thai and it looks great, and has good flavor in the sauce and tofu. Unfortunately, the dish is completely undermined by an abundance of chicken that is California drought DRY. For a $22 Pad Thai, probably the most expensive Chicken Pad Thai I've had by a long shot, this is just immensely disappointing, especially because it looked promising.
Osha has interestingly printed napkins with their logo in gold, which looks nice, but felt like sandpaper in practice.

As an expert salad maker, I have to say this is just a show-stopping salad! The dehydrated seaweed crisps add a subtle sweetness to the dish that is just wants you to dig into more. The wasabi peas are a good textural and flavor contrast to the salmon and rest of the dish to give it a kick. Also, the fresh tomato, sprouts, cucumber, and avocado are classic pairings that work well in a delicate salad. Overall, this is a beautiful plate of salad that's clearly crafted with intention, and tastes absolutely fantastic! I have never been this impressed by a salad in a long time, so kudos to Blue Owl!

A Simple Linear Model

* Approximate restaurant rating score y_i as $\sum_{i=1}^{d} w_i x_i + b$

> Q: what does a larger versus smaller weight w_i indicate about feature x_i?

A Simple Linear Model

* Approximate restaurant rating score y_i as $\sum_{i=1}^{d} w_i x_i + b$

> Q: what does a larger versus smaller weight w_i indicate about feature x_i?

* **Goal:** learn optimal weights w^* that best aligns restaurant score w.r.t. features

* Rewrite using homogeneous coordinates: $\sum_{i=0}^{d} w_i x_i = w^T x$

* Notice the formulation $y \sim w^T x$ is a linear regression model

> Q: how does the rewritten version integrate the bias term into vector notation?

A Simple Linear Model

* Approximate restaurant rating score y_i as $\sum_{i=1}^{d} w_i x_i + b$

> Q: what does a larger versus smaller weight w_i indicate about feature x_i?

* **Goal:** learn optimal weights w^* that best aligns restaurant score w.r.t. features

* Rewrite using homogeneous coordinates: $\sum_{i=0}^{d} w_i x_i = w^T x$

* Notice the formulation $y \sim w^T x$ is a linear regression model

> Q: how does the rewritten version integrate the bias term into vector notation?

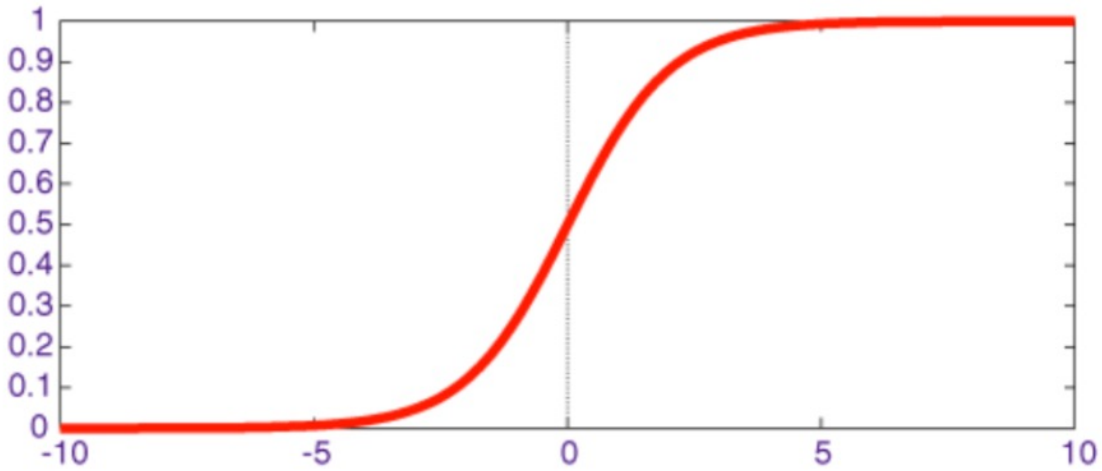
> Q: what if we want a probabilistic interpretation instead?

The Sigmoid Function

* Denote sigmoid as $\theta(s) = \frac{e^s}{1+e^s} \in (0,1)$

* Built-in probabilistic interpretation

* Property I: $\theta(s) = \frac{e^s}{1+e^s} = \frac{1}{1+e^{-s}}$



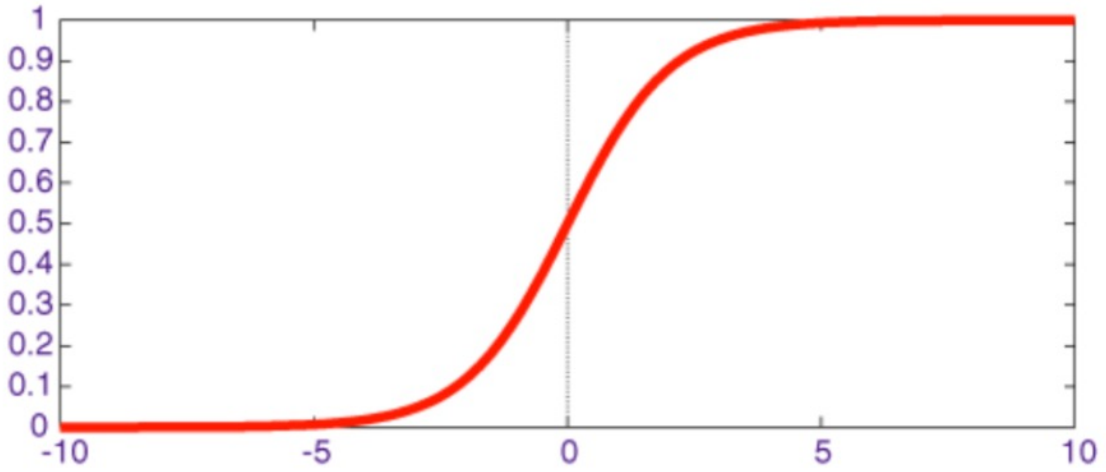
The Sigmoid Function

* Denote sigmoid as $\theta(s) = \frac{e^s}{1+e^s} \in (0,1)$

* Built-in probabilistic interpretation

* Property I: $\theta(s) = \frac{e^s}{1+e^s} = \frac{1}{1+e^{-s}}$

* Property II: $\theta(-s) = 1 - \theta(s)$



Logistic Classifier Formulation

* Input: features – $x \in R^d \equiv X$

* Output: binary label (good or bad restaurant) – $y \in \{-1, 1\} \equiv Y$

* Logistic Regression Model: $\theta(w^T x)$

> Q: how we can use the logistic regression model as a binary classifier?

Group Activity: Logistic Regression Objective

* Logistic Regression Model: $\theta(w^T x)$

* Logistic Loss Function $\mathrm{L}(w) = \frac{1}{n}\sum_{i=1}^{n} \ln(1 + \exp(-y_i w^T x_i))$

I. Introduce yourself to your neighbor and perhaps make a new friend in this class!

II. Then, spend several minutes to try to understand whether the formulation makes sense.

(a) What is the probabilistic interpretation of a given regression model $\theta(w^T x)$?

(i.e., How do we compute $P(y_i \mid x_i)$?)

(b) For $y_i = +1$, what does the regression model $\theta(w^T x)$ encourage? Does this make sense?

For $y_i = -1$, what does the regression model $\theta(w^T x)$ encourage? Does this make sense?

Deriving the Logistic Objective

* Assume $D : \{(x_1, y_1), \ldots, (x_n, y_n)\}$ is independently generated
* **Goal**: pick weights w to maximize likelihood of $P(y_1, \ldots, y_n | x_1, \ldots, x_n; w)$

Deriving the Logistic Objective

* Assume $D : \{(x_1, y_1), \ldots, (x_n, y_n)\}$ is independently generated

* **Goal**: pick weights w to maximize likelihood of $P(y_1, \ldots, y_n | x_1, \ldots, x_n; w)$

By independence: $\max P(y_1, \ldots, y_n | x_1, \ldots, x_n; w) = \max \prod_{i=1}^{n} P(y_i | x_i; w)$

Deriving the Logistic Objective

* Assume $D : \{(x_1, y_1), \ldots, (x_n, y_n)\}$ is independently generated

* Goal: pick weights w to maximize likelihood of $P(y_1, \ldots, y_n | x_1, \ldots, x_n; w)$

By independence: $$\max P(y_1, \ldots, y_n | x_1, \ldots, x_n; w) = \max \prod_{i=1}^{n} P(y_i | x_i; w)$$

Since log is monotonically increasing: $$= \max \ln\left(\prod_{i=1}^{n} P(y_i | x_i; w)\right)$$

Deriving the Logistic Objective

* Assume $D : \{(x_1, y_1), \ldots, (x_n, y_n)\}$ is independently generated

* **Goal**: pick weights w to maximize likelihood of $P(y_1, \ldots, y_n | x_1, \ldots, x_n; w)$

By independence: $\max P(y_1, \ldots, y_n | x_1, \ldots, x_n; w) = \max \prod_{i=1}^{n} P(y_i | x_i; w)$

Since log is monotonically increasing: $= \max \ln(\prod_{i=1}^{n} P(y_i | x_i; w))$

By log product rule: $= \max \sum_{i=1}^{n} \ln(P(y_i | x_i; w))$

Deriving the Logistic Objective

* Assume $D : \{(x_1, y_1), \ldots, (x_n, y_n)\}$ is independently generated

* Goal: pick weights w to maximize likelihood of $P(y_1, \ldots, y_n | x_1, \ldots, x_n; w)$

By independence: $\max P(y_1, \ldots, y_n | x_1, \ldots, x_n; w) = \max \prod_{i=1}^{n} P(y_i | x_i; w)$

Since log is monotonically increasing: $= \max \ln(\prod_{i=1}^{n} P(y_i | x_i; w))$

By log product rule: $= \max \sum_{i=1}^{n} \ln(P(y_i | x_i; w))$

Maximizing x is minimizing –x: $= \min \sum_{i=1}^{n} -\ln(P(y_i | x_i; w))$

Deriving the Logistic Objective

* Assume $D : \{(x_1, y_1), \ldots, (x_n, y_n)\}$ is independently generated

* Goal: pick weights w to maximize likelihood of $P(y_1, \ldots, y_n | x_1, \ldots, x_n; w)$

By independence: $\max P(y_1, \ldots, y_n | x_1, \ldots, x_n; w) = \max \prod_{i=1}^{n} P(y_i | x_i; w)$

Since log is monotonically increasing: $= \max \ln(\prod_{i=1}^{n} P(y_i | x_i; w))$

By log product rule: $= \max \sum_{i=1}^{n} \ln(P(y_i | x_i; w))$

Maximizing x is minimizing –x: $= \min \sum_{i=1}^{n} -\ln(P(y_i | x_i; w))$

By log power rule: $= \min \sum_{i=1}^{n} \ln(\frac{1}{P(y_i | x_i; w)})$

Deriving the Logistic Objective

This is an example of how to clearly and concisely show your work! Do not write long essays for math questions! Assignments are graded by both recall AND precision.

* Assume $D : \{(x_1, y_1), \dots, (x_n, y_n)\}$ is independently generated

* **Goal:** pick weights w to maximize likelihood of $P(y_1, \dots, y_n | x_1, \dots, x_n; w)$

By independence: $$\max P(y_1, \dots, y_n | x_1, \dots, x_n; w) = \max \prod_{i=1}^{n} P(y_i | x_i; w)$$

Since log is monotonically increasing: $$= \max \ln(\prod_{i=1}^{n} P(y_i | x_i; w))$$

By log product rule: $$= \max \sum_{i=1}^{n} \ln(P(y_i | x_i; w))$$

Maximizing x is minimizing –x: $$= \min \sum_{i=1}^{n} -\ln(P(y_i | x_i; w))$$

By log power rule: $$= \min \sum_{i=1}^{n} \ln(\frac{1}{P(y_i | x_i; w)})$$

Substitute probabilistic interpretation: $$= \min \sum_{i=1}^{n} \ln(\frac{1}{\theta(y_i w^T x_i)}) = \min \sum_{i=1}^{n} \ln(1 + \exp(-y_i w^T x_i))$$

Do We Need An Objective?

Recall from Linear Regression

* Objective: $\mathbf{L(w)} = ||Xw - y||_2{}^2$
* Gradient: $\nabla_w L(w) = X^T X w - X^T y$
* Closed Form Solution: $w^* = (X^T X)^{-1} X^T y$

Do We Need An Objective?

Recall from Linear Regression

* Objective: $\mathbf{L(w)} = ||Xw - y||_2{}^2$

* Gradient: $\nabla_w L(w) = X^T X w - X^T y$

* Closed Form Solution: $w^* = (X^T X)^{-1} X^T y$

For Logistic Regression

* Objective: $\mathbf{L}(w) = \frac{1}{n}\sum_{i=1}^{n} \ln(1 + \exp(-y_i w^T x_i))$

* Closed-Form Solution: unfortunately, none exists ._.

* Instead, we must use a variation of gradient descent!

Gradient Descent Algorithm

1. Initialize at step $t = 0$ to $\mathbf{w}(0)$.
2. **for** $t = 0, 1, 2, \ldots$ **do**
    1. Compute the gradient

$$\mathbf{g}_t = \nabla E(\mathbf{w}(t))$$

    2. Move in the direction $\mathbf{v}_t = -\mathbf{g}_t$
    3. Update the weights:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta \mathbf{v}_t$$

    4. Iterate 'until it is time to stop'
3. **end for**
4. Return the final weights.

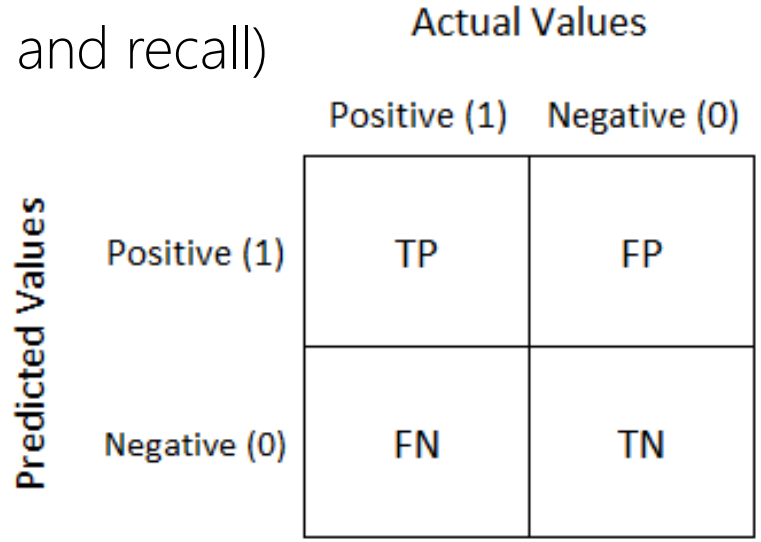
Performance Measures

* Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$

* Precision: $\frac{TP}{TP+FP}$ (maximizing precision is minimizing false alarms)

* Recall: $\frac{TP}{TP+FN}$ (maximizing recall is minimizing overlooked cases)

* F1 Score: $\frac{2*Precision*Recall}{Precision+Recall}$ (harmonic balance between precision and recall)



Changing the Decision Threshold

* If greater than threshold, classify positive; else, classify negative

* Default threshold: midpoint of range; for sigmoid activated logistic regression choose 0.5

> Q: what if we want to maximize precision?

> Q: what if we want to maximize recall?

At-Home Exercise

1. (25 points) Cross-entropy error measure.

   (a) (12 points) If we are learning from ± 1 data to predict a noisy target $P(y|\mathbf{x})$ with candidate hypothesis h, show that the maximum likelihood method reduces to the task of finding h that minimizes

$$E_{in}(\mathbf{w}) = \sum_{n=1}^{N} [\![y_n = +1]\!] \ln \frac{1}{h(\mathbf{x}_n)} + [\![y_n = -1]\!] \ln \frac{1}{1 - h(\mathbf{x}_n)}$$

   Hint: Use the likelihood $p(y|x) = \begin{cases} h(x) & \text{for } y = +1 \\ 1 - h(x) & \text{for } y = -1 \end{cases}$ and derive the maximum likelihood formulation.

   (b) (13 points) For the case $h(\mathbf{x}) = \theta\left(\mathbf{w}^T \mathbf{x}\right)$, argue that minimizing the in-sample error in part (a) is equivalent to minimizing the one given below

$$E_{in}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \ln\left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}\right)$$

   Note: For two probability distributions $\{p, 1-p\}$ and $\{q, 1-q\}$ with binary outcomes, the cross-entropy (from information theory) is

$$p \log \frac{1}{q} + (1-p) \log \frac{1}{1-q}.$$

   The in-sample error in part (a) corresponds to a cross-entropy error measure on the data point $(\mathbf{x}_n, y_n)$, with $p = [\![y_n = +1]\!]$ and $q = h(\mathbf{x}_n)$.

HW1 Talk

Professor: Please expand $\frac{1}{2}\nabla(\mathbf{A}\cdot\mathbf{A})$

The nerdy kid who shouts

$(\mathbf{A}\cdot\nabla)\mathbf{A} + \mathbf{A}\times(\nabla\times\mathbf{A})$

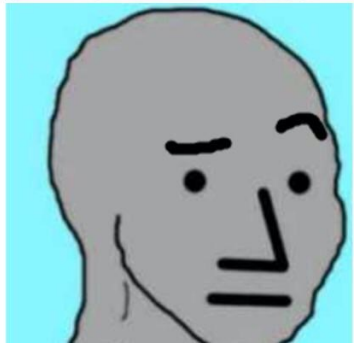
The quiet kid who whispers

$\mathbf{A}\cdot\mathbf{J}_\mathbf{A}$

Me who doesn't even understand what

$(\mathbf{A}\cdot\nabla)\mathbf{A}$

means or how it works



Acknowledgement

Many slides are adapted from Professor Shiyu Chang at UCSB:

https://code-terminator.github.io/