Logistic Regression and Gradient Descent

Lecture 4

Last Time

- ***Linear regression*** is a hypothesis class for predicting a continuous value from a linear combination of the attributes

- ***Polynomial regression*** generalizes linear regression. Same loss and optimizer, but we expand representation by taking polynomials of chosen degree of each attribute (still linear in weights so same algorithm!)

- Textbook: section 9.2

This Class

- How can we build linear predictors
  - for predicting probabilities of discrete classes?
  - for predicting more than two classes?
- Textbook: Chapters 9.3, 12.1.1, 14.0, 14.1.0, 14.3.0, 14.5.1

Motivation

Predicting Probabilities

- We can learn hypotheses that output probabilities of discrete labels, not just the labels themselves

- What applications can you think of where this is an advantage?

p(cute) = 99%

p(cute) = 45%

p(cute) = 1%

Logistic Regression

Logistic Regression

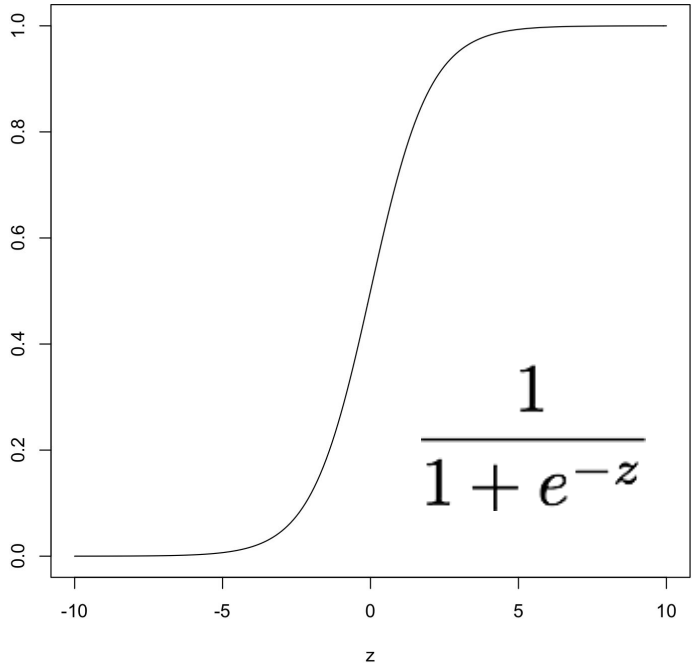
Logistic regression is another common hypothesis class, again for classification

$$\mathcal{X} = \mathbb{R}^d \quad \mathcal{Y} = \{1, -1\}$$

Now we use a linear predictor that outputs a continuous value in [0, 1]

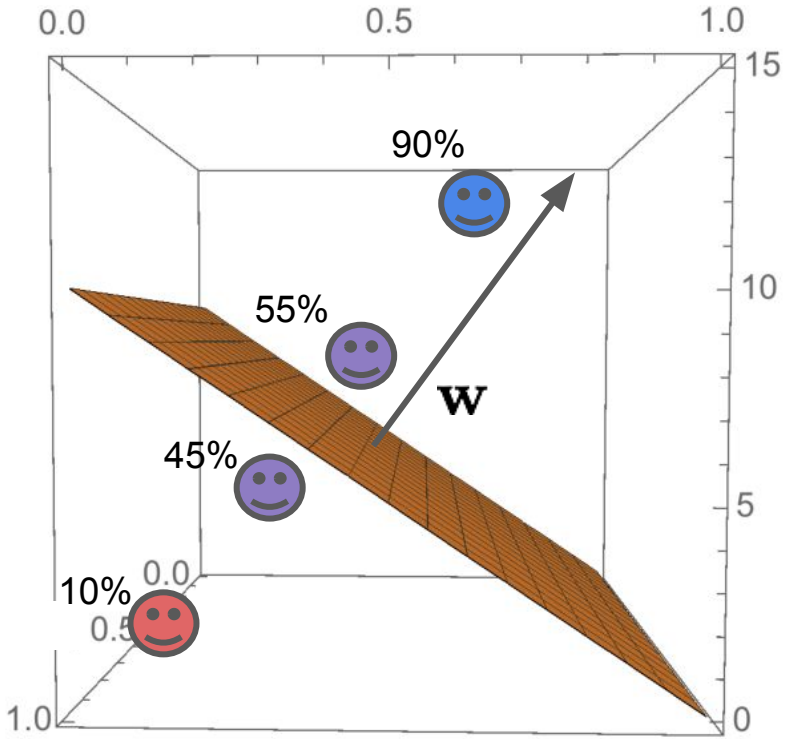
$$h_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1 + e^{-\langle \mathbf{w}, \mathbf{x} \rangle}}$$

$$h : \mathcal{X} \to [0, 1]$$



Logistic Regression Decision Boundary

- Decision boundary is still solution to $\langle \mathbf{w}, \mathbf{x} \rangle = 0$
- Corresponds to predicted probability of each class is 50%
- Probability changes with distance from decision boundary



Multiclass Logistic Regression

Multiclass Logistic Regression

What if we want to predict from among more than two classes?

$$\mathcal{X} = \mathbb{R}^d \quad \mathcal{Y} = \{1, \ldots, k\}$$

Now we have a vector-valued function where each component j is:

$$h_{\mathbf{w}}(\mathbf{x})_j = \frac{e^{\langle \mathbf{w}_j, \mathbf{x} \rangle}}{\sum_{s=1}^{k} e^{\langle \mathbf{w}_s, \mathbf{x} \rangle}}$$

$h : \mathcal{X} \to [0,1]^k \qquad \mathbf{w} \in \mathbb{R}^{k \times d}$

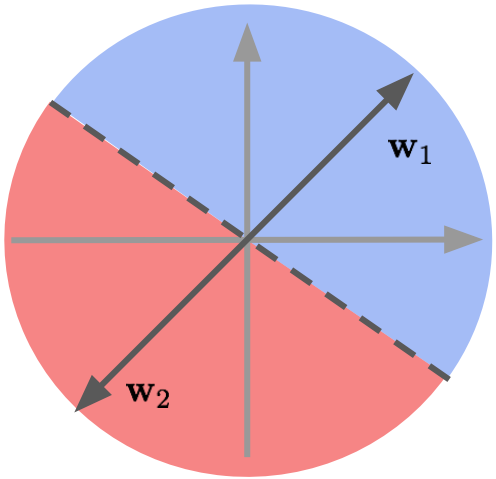
$$\begin{pmatrix} h_{\mathbf{w}}(\mathbf{x})_1 \\ \vdots \\ h_{\mathbf{w}}(\mathbf{x})_k \end{pmatrix}$$

is also called the softmax of

$$\begin{pmatrix} \langle \mathbf{w}_1, \mathbf{x} \rangle \\ \vdots \\ \langle \mathbf{w}_k, \mathbf{x} \rangle \end{pmatrix}$$

Binary vs Multiclass Logistic Regression

1. Start with “multiclass” with 2 classes



2. Constrain

$$\mathbf{w}_2 = \mathbf{0}$$

3. Simplify

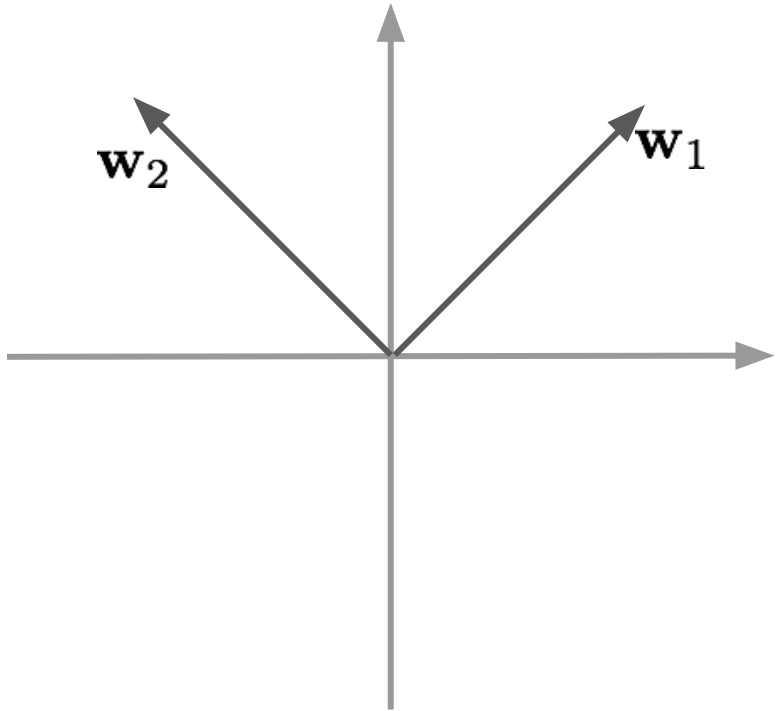
$$h_{\mathbf{w}}(\mathbf{x})_1 = \frac{e^{\langle \mathbf{w}_1, \mathbf{x} \rangle}}{e^{\langle \mathbf{w}_1, \mathbf{x} \rangle} + e^{\langle \mathbf{w}_2, \mathbf{x} \rangle}}$$

$$= \frac{e^{\langle \mathbf{w}_1, \mathbf{x} \rangle}}{e^{\langle \mathbf{w}_1, \mathbf{x} \rangle} + e^{0}}$$

$$= \frac{1}{1 + e^{-\langle \mathbf{w}_1, \mathbf{x} \rangle}}$$

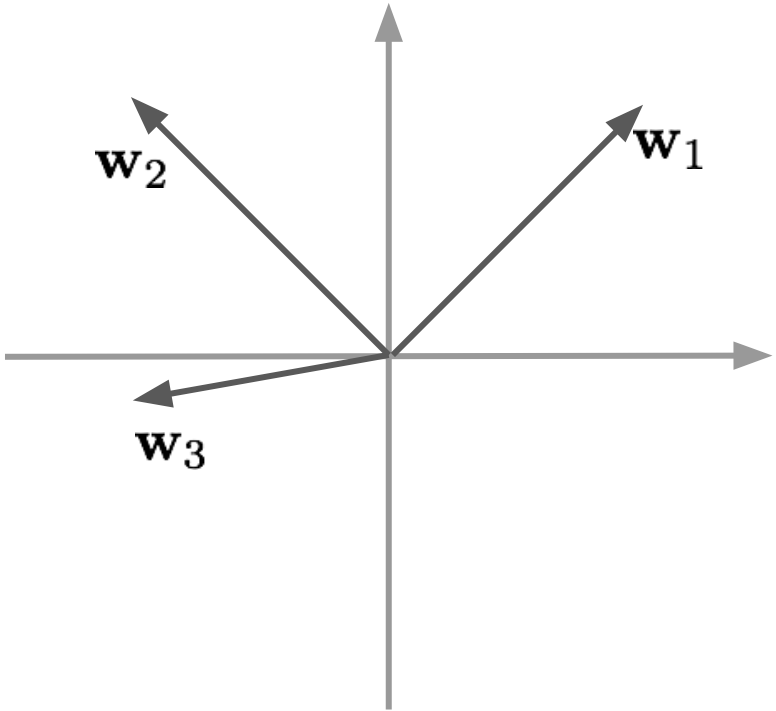
Example: 2 Classes in 2 Dimensions

$$\mathcal{X} = \mathbb{R}^2$$
$$\mathcal{Y} = \{1, 2\}$$



Example: 3 Classes in 2 Dimensions

$$\mathcal{X} = \mathbb{R}^2$$
$$\mathcal{Y} = \{1, 2, 3\}$$



Question

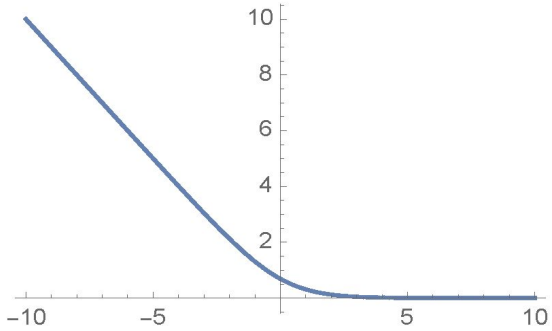
We’ll Have to Wait for the Question

Log Loss

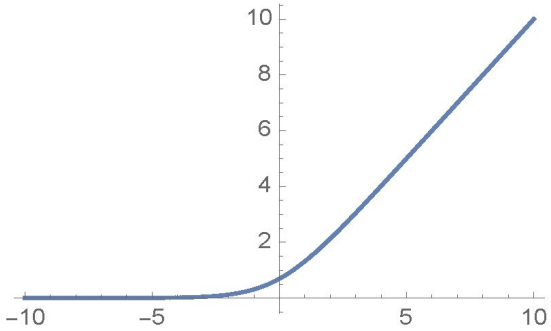
Binary Log Loss

Learning minimizes a new loss function, called the log loss (or logistic loss), that penalizes degree of wrongness:

$$\ell(h_{\mathbf{w}}, (\mathbf{x}, y)) = \log(1 + \exp(-y\langle \mathbf{w}, \mathbf{x} \rangle))$$



$y = 1$



$y = -1$

Multiclass Log Loss

Since our hypothesis outputs a vector of values, need to consider them all:

$$\ell(h_{\mathbf{w}}, (\mathbf{x}, y)) = -\sum_{j=1}^{k} \mathbf{1}[y = j] \log h_{\mathbf{w}}(\mathbf{x})_j$$

$$L_S(h_{\mathbf{w}}) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{k} \mathbf{1}[y_i = j] \log h_{\mathbf{w}}(\mathbf{x}_i)_j$$

Also called the cross entropy loss

Optimizing the Multiclass Log Loss

Now we need to search for a k x d weight matrix $\mathbf{W}$.

Can we find one where the gradient of the empirical risk is zero?

First Step: Gradient of Softmax

Recall that our predicted probability that example i has class j is:

$$h_{\mathbf{w}}(\mathbf{x}_i)_j = \frac{e^{\langle \mathbf{w}_j, \mathbf{x}_i \rangle}}{\sum_{s=1}^{k} e^{\langle \mathbf{w}_s, \mathbf{x}_i \rangle}}$$

Then:

$$\frac{\partial h_{\mathbf{w}}(\mathbf{x}_i)_j}{\partial w_{st}} = \begin{cases} -h_{\mathbf{w}}(\mathbf{x}_i)_j \cdot h_{\mathbf{w}}(\mathbf{x}_i)_s \cdot x_{it} & \text{if } j \neq s \\ 0 & \text{if } j = s \end{cases}$$

$s \in [k]$
$t \in [d]$

Optimizing the Multiclass Log Loss

$$L_S(h_{\mathbf{w}}) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}\mathbf{1}[y_i = j]\log h_{\mathbf{w}}(\mathbf{x}_i)_j$$

$$\frac{\partial L_S(h_{\mathbf{w}})}{\partial w_{st}} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}\mathbf{1}[y_i = j]\frac{1}{h_{\mathbf{w}}(\mathbf{x}_i)_j}\frac{\partial h_{\mathbf{w}}(\mathbf{x}_i)_j}{\partial w_{st}}$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}\mathbf{1}[y_i = j \wedge j \neq s](-h_{\mathbf{w}}(\mathbf{x}_i)_s \cdot x_{it})$$

$$= \frac{1}{m}\sum_{i=1}^{m}(h_{\mathbf{w}}(\mathbf{x}_i)_s - \mathbf{1}[y_i = s])x_{it}$$

$s \in [k]$
$t \in [d]$

Optimizing the Multiclass Log Loss

$$\frac{\partial L_S(h_{\mathbf{w}})}{\partial w_{st}} = \frac{1}{m}\sum_{i=1}^{m}(h_{\mathbf{w}}(\mathbf{x}_i)_s - \mathbf{1}[y_i = s])x_{it}$$

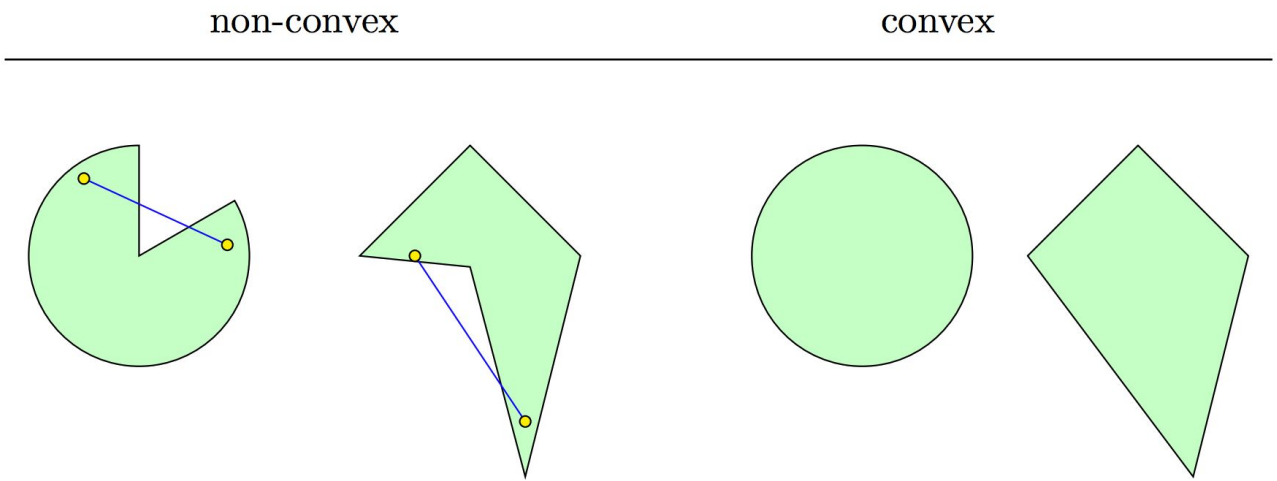
$$= 0$$

We don’t have a closed form solution for $\mathbf{w}$!

$s \in [k]$
$t \in [d]$

Convexity

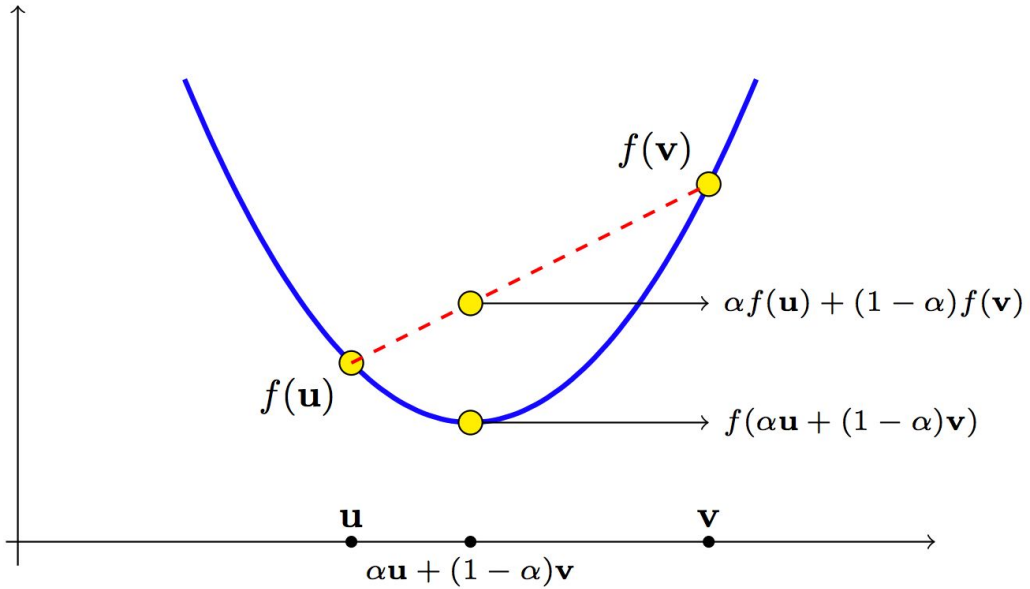
Convex Set



DEFINITION 12.1 (Convex Set) A set C in a vector space is convex if for any two vectors $\mathbf{u}, \mathbf{v}$ in C, the line segment between $\mathbf{u}$ and $\mathbf{v}$ is contained in C. That is, for any $\alpha \in [0, 1]$ we have that $\alpha\mathbf{u} + (1 - \alpha)\mathbf{v} \in C$.

Understanding Machine Learning.
Shalev-Shwartz and Ben-David, 2014.

Convex Function



DEFINITION 12.2 (Convex Function) Let C be a convex set. A function $f : C \to \mathbb{R}$ is convex if for every $\mathbf{u}, \mathbf{v} \in C$ and $\alpha \in [0, 1]$,

$$f(\alpha\mathbf{u} + (1-\alpha)\mathbf{v}) \leq \alpha f(\mathbf{u}) + (1-\alpha) f(\mathbf{v}) .$$

Understanding Machine Learning. Shalev-Shwartz and Ben-David, 2014.

Proving that the Log Loss is Convex

The log loss (binary or multiclass) is convex in the weights:

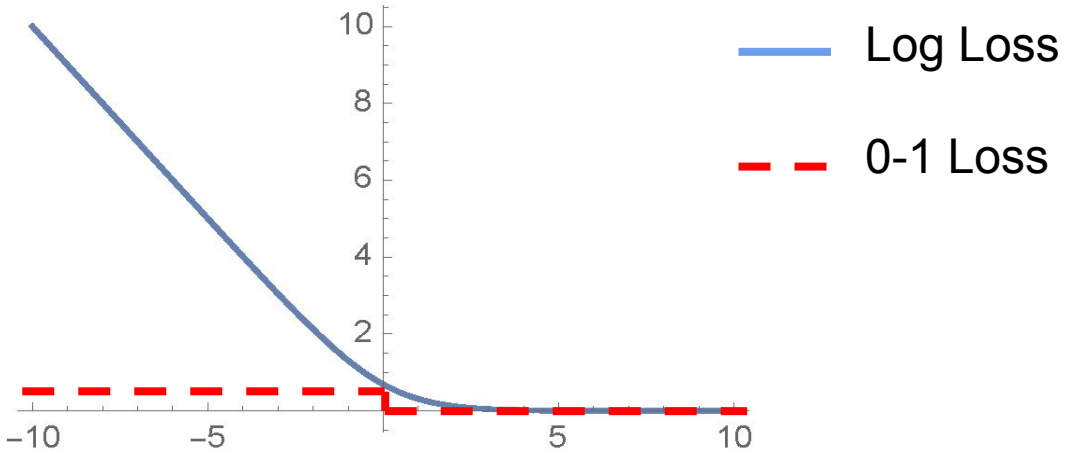
1. $\mathbb{R}^{k \times d}$ is a convex set

2. Log loss for one example is convex in the weights
   a. Option 1: The second derivative test
   b. Option 2: Line test

3. $L_S(h_{\mathbf{w}})$ is convex in the weights
   a. Sum of convex functions with nonnegative weights is convex

Log Loss vs 0-1 Loss

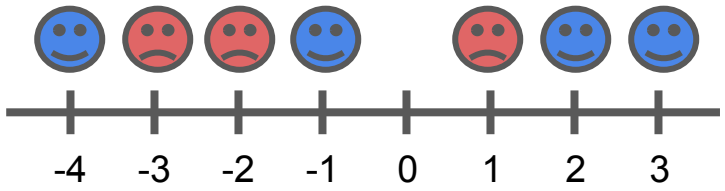
- Comparing the binary log loss to the 0-1 loss gives an intuition:



- Log loss is convex upper bound on 0-1 loss!

Example: Log Loss vs 0-1 Loss

- Suppose we have this 1-d data:

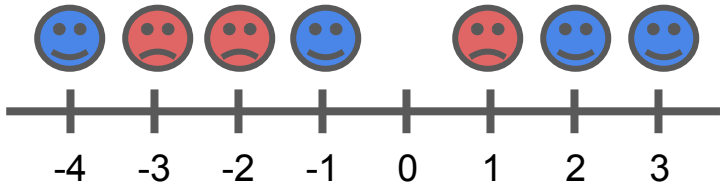


- And suppose we want to fit a halfspace of the form $\text{sign}[w \cdot x + 1]$

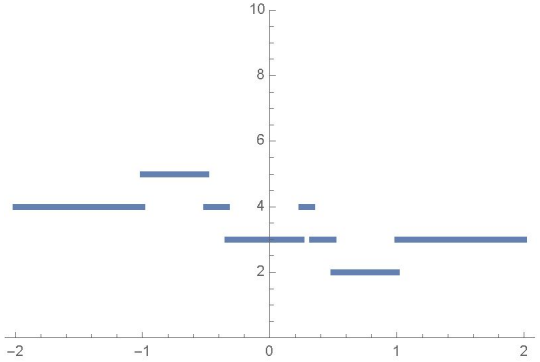
- Then the ERM for 0-1 loss is $w \in (0.5, 1.0)$

Example: Log Loss vs 0-1 Loss

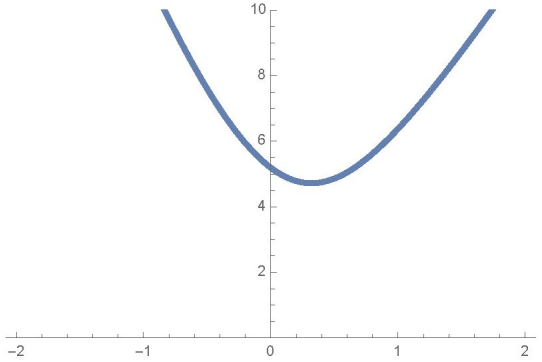
- Suppose we have this 1-d data:



- Loss functions:



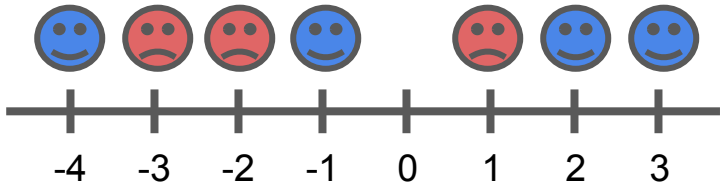
0-1 Loss for Halfspace



Log Loss for Logistic Regression

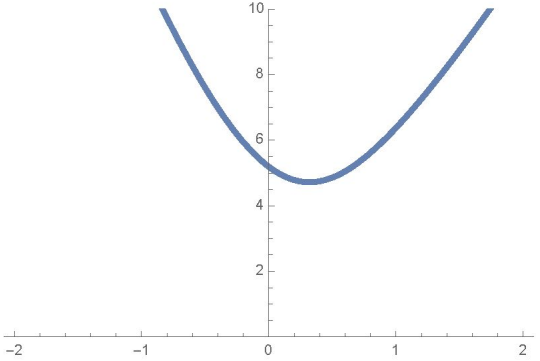
Example: Log Loss vs 0-1 Loss

- Suppose we have this 1-d data:



- Minimizer for log loss is ~.32

- Decision boundary is ~-3.1



Gradient Descent

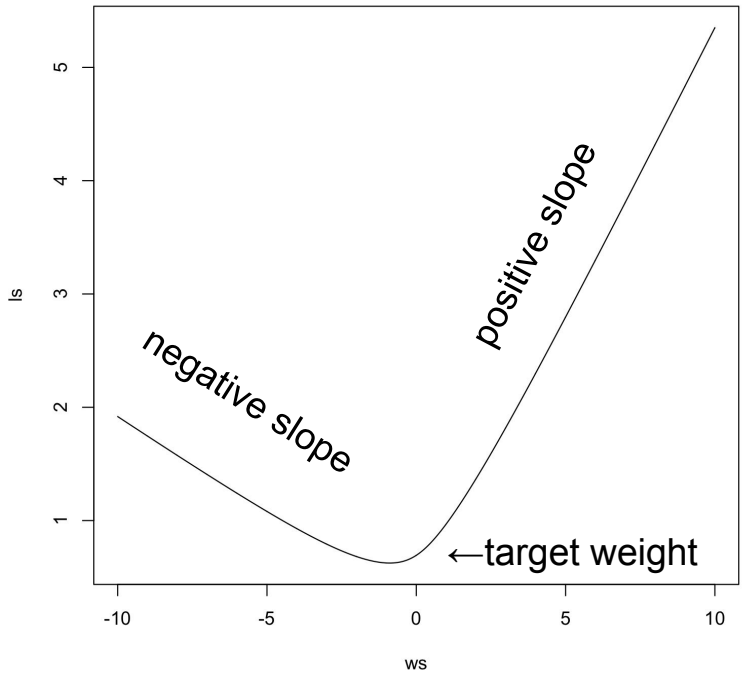
Gradient Descent

Our function has a minimum and the derivatives point to it.

$$w_{st} \leftarrow w_{st} - \alpha \frac{\partial L_S(h_{\mathbf{w}})}{\partial w_{st}}$$

α is a user-specified value, i.e., a hyperparameter, called the step size

Repeat the update until convergence, i.e., the loss doesn't change significantly



Question

We’ll Have to Wait for the Question

How do we Set the Step Size?

- Since the loss for logistic regression is convex, okay to try different values and see which leads to lowest training loss in reasonable time.

- Common values range from $1e^{-3}$ to $1e^{-6}$

- Some optimizers automatically tune step size, such as AdaGrad and Adam

Gradient Descent in Higher Dimensions

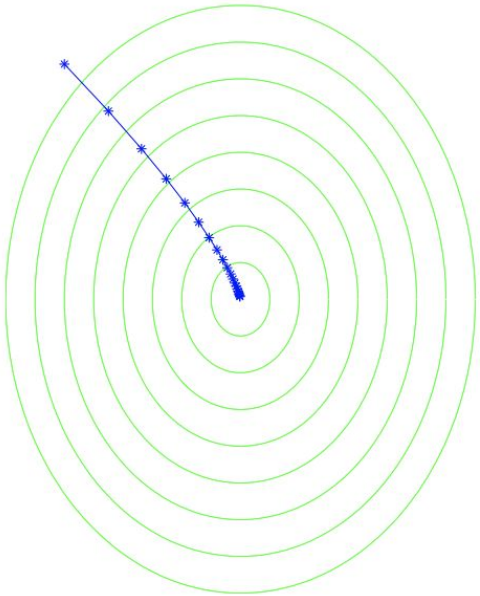


Figure 14.1 An illustration of the gradient descent algorithm. The function to be minimized is $1.25(x_1 + 6)^2 + (x_2 - 8)^2$.

Understanding Machine Learning.
Shalev-Shwartz and Ben-David, 2014.

The Most Important Things

- ***Logistic regression*** is a hypothesis class for predicting the probability of a discrete class label, which is log proportional to linear combination of attributes
  - We can generalize it to the multiclass case (k > 2)
  - But no closed-form solution for ERM

- To find ERM of a hypothesis class like multiclass logistic regression, we can use ***gradient descent***: repeatedly change the weights in the opposite direction of the gradient of the risk

- Textbook: sections 9.3, 12.1.1, 14.0, 14.1.0

Next Class

- How do we apply logistic regression to real-world problems that have
  - Lots of data?
  - Features of different types?
- Textbook: sections 14.3.0, 14.5.1