Halfspaces and Perceptron

Lecture 2

Last Time

- We focus on supervised machine learning, with some unsupervised in April
- We use ***empirical risk minimization (ERM)***
    - ERM = pick a hypothesis that minimizes the loss (i.e. empirical risk) on a set of training data
- Naively applying ERM can lead to the pitfall of ***overfitting***
    - Overfitting = picking a hypothesis that is great on training data but very bad on new test data
- Textbook: chapter 1, sections 2.0, 2.1, 2.2

This Class

- What is a practically useful class of hypotheses that often avoids overfitting?
- How to select an ERM hypothesis from that class computationally efficiently?
- Textbook: sections 9.0, 9.1.0, 9.1.2

Example Revisited:
Does this animal have cute babies?

Example Training Data

Tiger

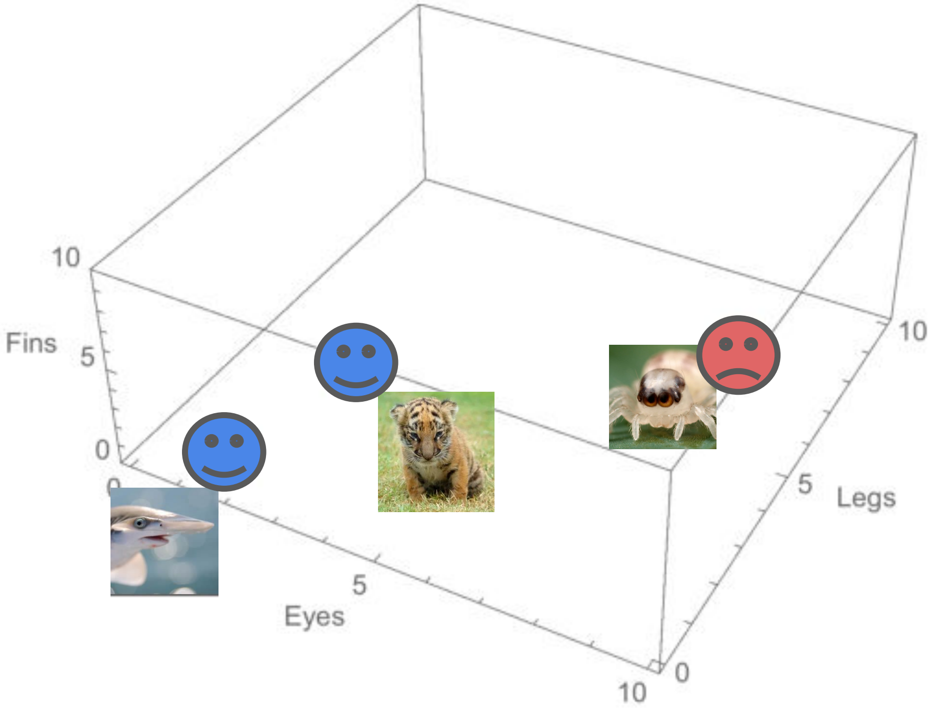
$$z_1 = (2, 4, 0, 1)$$

Spider

$$z_2 = (8, 8, 0, -1)$$

Shark

$$z_3 = (2, 0, 2, 1)$$

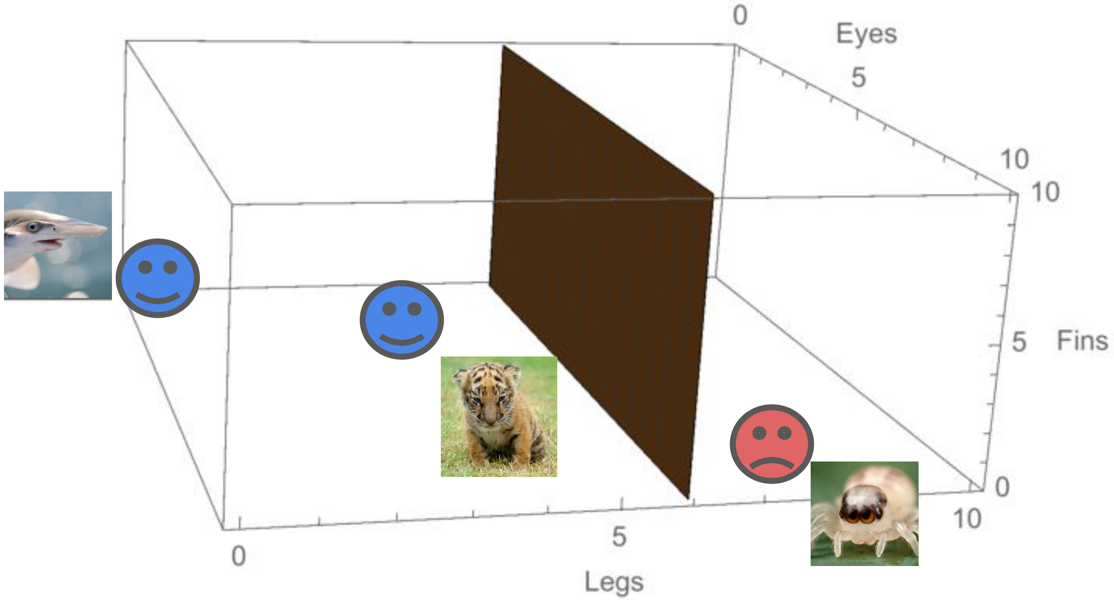


Example hypothesis

h = If legs <= 6, then cute
Otherwise, not cute

Equivalent to:

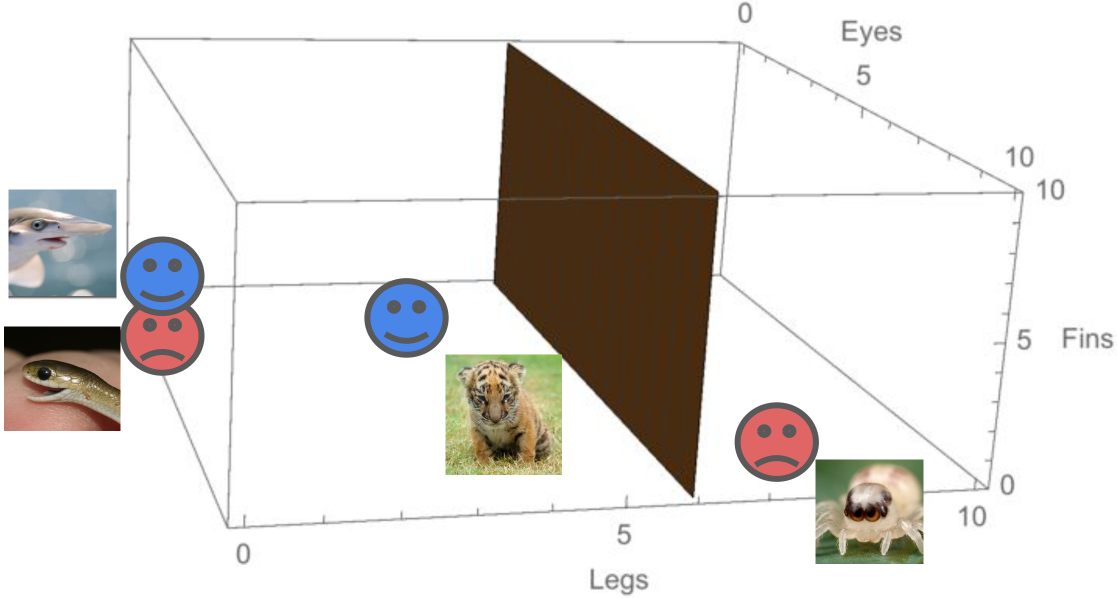
If -1 * legs + 6 > 0, then cute
Otherwise, not cute



What if we also had Snake?

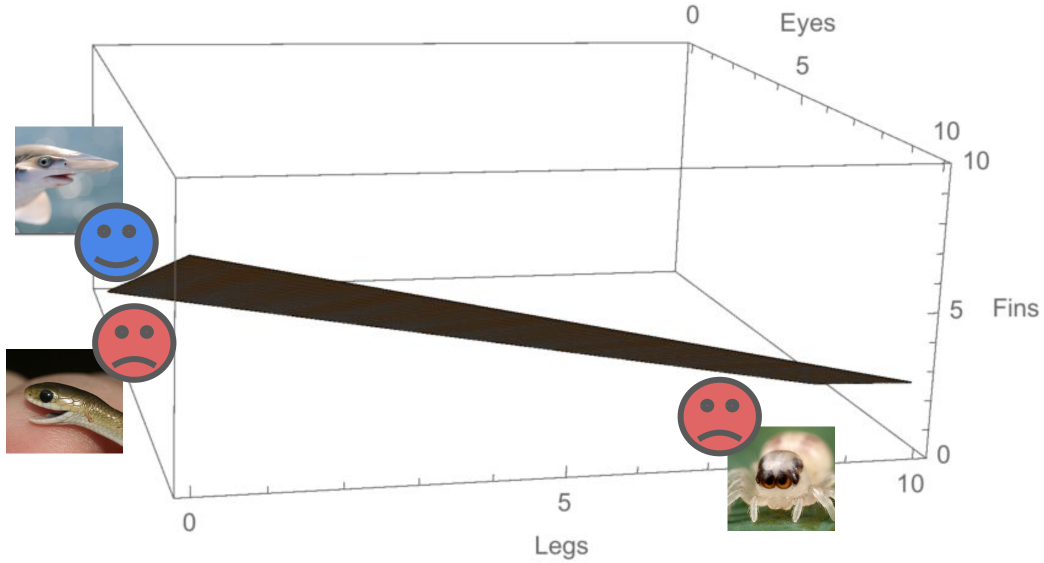
Snake

$$z_4 = (2, 0, 0, -1)$$



Example improved hypothesis

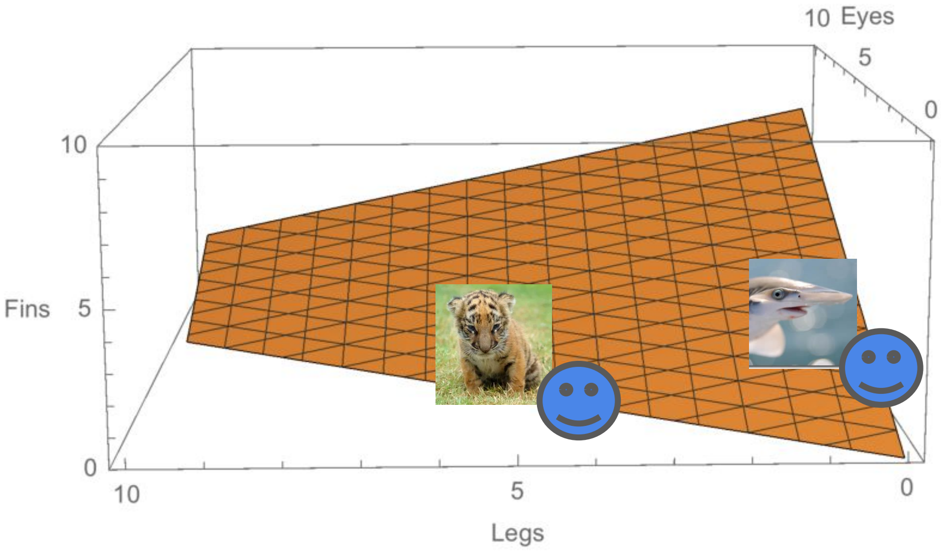
h = If -1.5 * Eyes + Legs + 2 * Fins > 0, then cute
Otherwise, not cute



Alternate View (Rotated Horizontally 180°)

h = If -1.5 * Eyes + Legs + 2 * Fins >= 0, then cute

Otherwise, not cute



Halfspaces

Affine Function

Fancy name for a linear function plus a constant:

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

where $\mathbf{x} \in \mathbb{R}^d, b \in \mathbb{R}$

Often, we will omit b and assume that the last element of each $\mathbf{x}$ is 1

Halfspace Hypothesis

We can use an affine function to define a hypothesis called a halfspace:

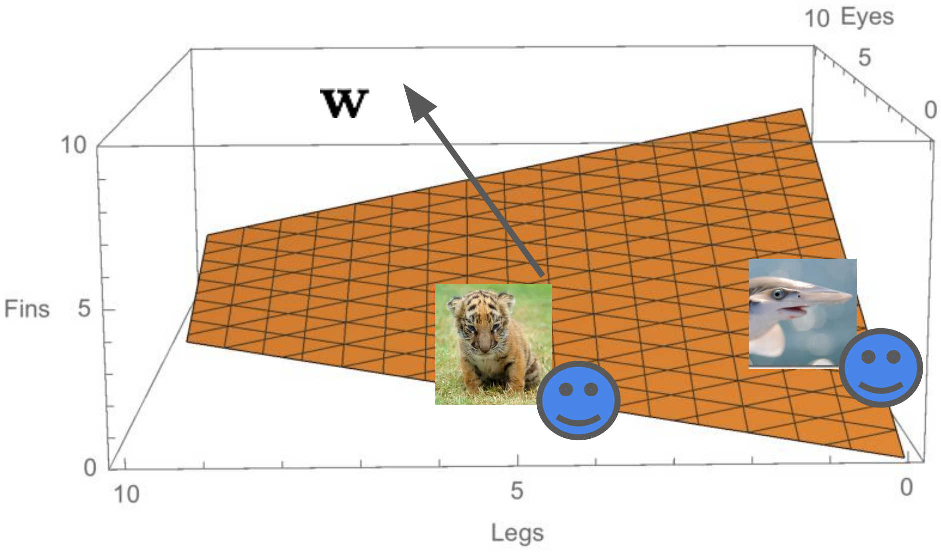
$$h_{\mathbf{w}}(\mathbf{x}) = \operatorname{sign}(\langle \mathbf{w}, \mathbf{x} \rangle)$$

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

Decision Boundary

In a d-dimensional attribute space the decision boundary is the (d-1)-dimensional hyperplane where $\langle \mathbf{w}, \mathbf{x} \rangle = 0$

$\mathbf{w}$ is a vector normal to that hyperplane (pointing towards the positive side)



Question

How We’ll Do Questions: Peer Instruction

1. Solo Vote
2. Group Discussion
3. Group Vote
4. Class Discussion

Remember: You get full credit for participating, not selecting the correct answer

And backed by SCIENCE!

A Multi-institutional Study of Peer Instruction in Introductory Computing

Leo Porter, Dennis Bouvier, Quintin Cutts, Scott Grissom, Cynthia Lee, Robert McCartney, Daniel Zingaro, Beth Simon.

SIGCSE 2016, March 2016.

We’ll Have to Wait for the Question

Linear Separation and Realizability

Linear Separation

- What types of training data can we separate perfectly with a halfspace?
- I.e., when does there exist $\mathbf{w}$ such that $L_{\mathcal{D}}(h_{\mathbf{w}}) = 0$?
- More generally, beyond halfspace classifiers, called the **realizability assumption**
- Note that the realizability assumption implies $L_{S}(h_{\mathbf{w}}) = 0$ for all $S \sim \mathcal{D}$

Example

Let $f : \mathcal{X} \to \mathcal{Y}$ be the true labeling function, i.e., $\mathcal{D}(y = f(\mathbf{x})|\mathbf{x}) = 1$

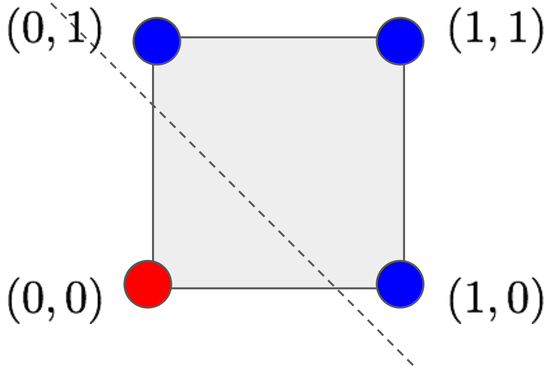
(and let $\mathcal{D}(\mathbf{x})$ be uniform)

Linearly Separable!

Suppose $\mathcal{X} = \{0,1\}^d, \quad \mathcal{Y} = \{1,-1\},$

and

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{i=1}^{d} x_i \geq 1 \\ -1 & \text{otherwise} \end{cases}$$



Question

We’ll Have to Wait for the Question

Perceptron Algorithm

Intuition

- Iterative algorithm
- Initialize weights to be all zeroes
- For each misclassified point, add the point (signed by its label) to the weight vector to shift it
- Continue until all points are classified correctly (or stuck)

Perceptron Training Algorithm

$w^{(1)} = (0, 0, \ldots, 0)$
for $t = 1, 2, \ldots$
    if $(\exists i \text{ s.t. } y_i \langle w, x_i \rangle \leq 0)$
        $w^{(t+1)} = w^{(t)} + y_i x_i$
    else
        return $w^{(t)}$

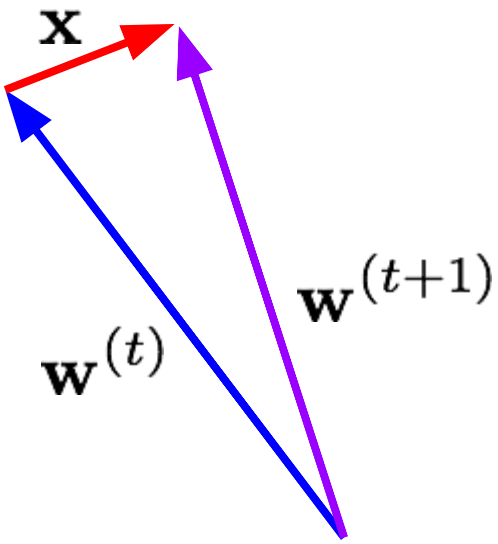
Animation Example

https://www.youtube.com/watch?v=xpJHhHwR4DQ

Why Does it Work?

- Halfspace is correct if $y\langle \mathbf{w}, \mathbf{x} \rangle > 0$
- Suppose for some training example that $y\langle \mathbf{w}, \mathbf{x} \rangle \leq 0$
- Adding $y\mathbf{x}$ to $\mathbf{w}$ moves weights toward right direction
- Always helps:

$$\begin{aligned} y\langle \mathbf{w}^{(t+1)}, \mathbf{x} \rangle &= y\langle \mathbf{w}^{(t)} + y\mathbf{x}, \mathbf{x} \rangle \\ &= y\langle \mathbf{w}^{(t)}, \mathbf{x} \rangle + \|\mathbf{x}\|_2^2 \end{aligned}$$



Formal Guarantee

THEOREM 9.1 *Assume that* $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ *is separable, let* $B = \min\{\|\mathbf{w}\| : \forall i \in [m],\ \ y_i\langle \mathbf{w}, \mathbf{x}_i\rangle \geq 1\}$, *and let* $R = \max_i \|\mathbf{x}_i\|$. *Then, the Perceptron algorithm stops after at most* $(RB)^2$ *iterations, and when it stops it holds that* $\forall i \in [m],\ \ y_i\langle \mathbf{w}^{(t)}, \mathbf{x}_i\rangle > 0$.

Proof sketch: let $\mathbf{w}^\star$ be the weight vector that achieves the min. in the definition of B and show that after T iterations:

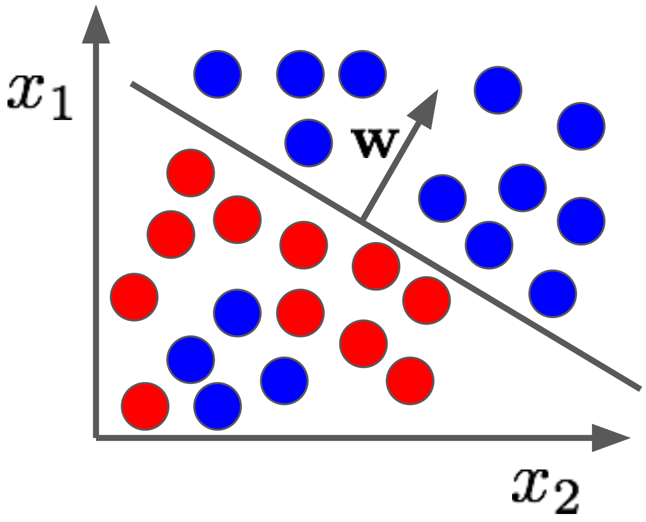
$$\frac{\langle \mathbf{w}^\star, \mathbf{w}^{(T+1)}\rangle}{\|\mathbf{w}^\star\|\ \|\mathbf{w}^{(T+1)}\|} \geq \frac{\sqrt{T}}{RB}$$

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

Do Halfspaces Ever Overfit?

Halfspaces seem pretty great, right?

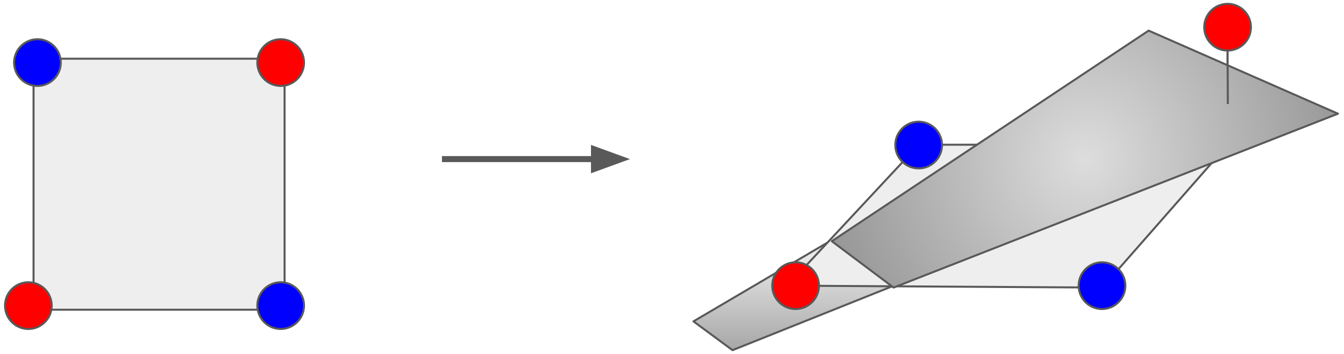
- If the data is linearly separable, then it's awesome
- If it's not, we can get something like this:



More Dimensions can Help

- Think back to the xor labeling function: $f^{\mathrm{C}}(\mathbf{x}) = x_1 \text{ xor } x_2$

- We could make such a 2-d data set into a 3-d linearly separable data set through feature engineering, i.e., expanding $\mathcal{H}$ to operate on new attributes $\mathbf{x}' = \psi(\mathbf{x})$ where $\psi(\mathbf{x}) = (x_1,\ x_2,\ x_1 \cdot x_2)$



Too Many Dimensions can Lead to Overfitting

- We can place d+1 points in $\mathbb{R}^d$ and linearly separate **any** labels assigned to them

- Example: if our points are $\mathbf{0}, \mathbf{e}_1 = (1, 0, 0, \ldots), \mathbf{e}_2 = (0, 1, 0, \ldots), \ldots$ then for any $y_1, y_2, \ldots$, set b to y_1 and $\mathbf{w}$ to $y_2, y_3, \ldots$

- The technical term for this is ***shattering***

So How Do We Know What to Do?

- Most of supervised machine learning is balancing the "triple tradeoff" (Dietterich, 2003) of the complexity of $\mathcal{H}$, the size of the training data m , and the true error $L_{\mathcal{D}}(h_S)$

- Throughout the semester we'll see formally and in practice that we need domain knowledge to balance this triple tradeoff. There is no free lunch!

Our first encounter with the curse of dimensionality...

I’ll just leave this here: https://www.youtube.com/watch?v=DQWI1kvmwRg

The Most Important Things

- ***Halfspaces*** are hypotheses defined as hyperplanes that separate two classes
- Training data that can be separated perfectly is ***linearly separable***
    - More general term for any hypothesis class with perfect fit for training data: ***realizable***
- The ***perceptron*** algorithm is a simple method for learning halfspaces
    - Finds ERM solution efficiently if training data is linearly separable
- Textbook: sections 9.0, 9.1.0, 9.1.2

Next Class

- How can we build linear predictors for predicting continuous values?
- Textbook: sections 9.2