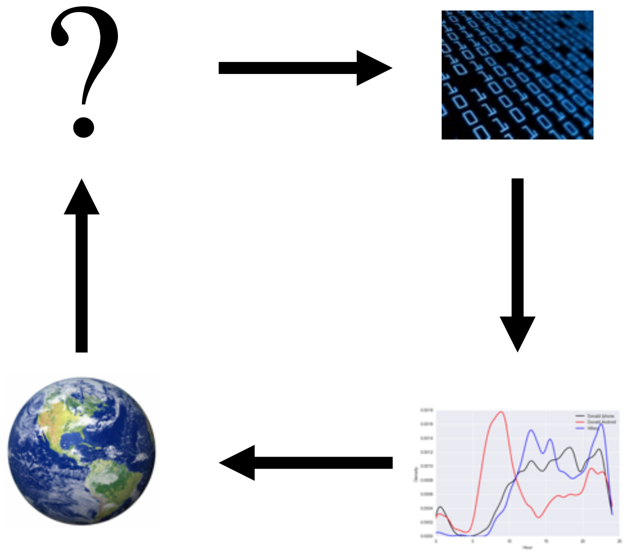
Classification
& Logistic Regression
& maybe deep learning

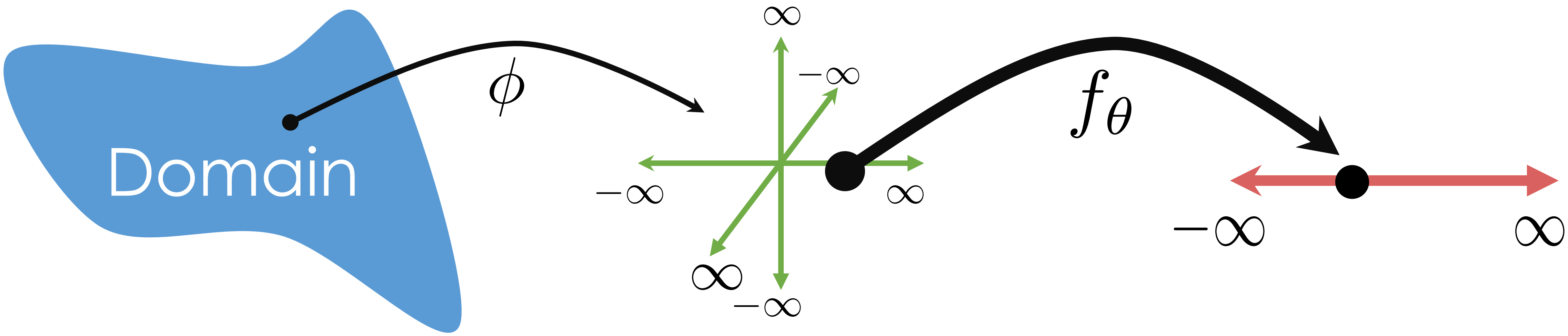
Slides by:

Joseph E. Gonzalez jegonzal@cs.berkeley.edu



Previously…

So far ....

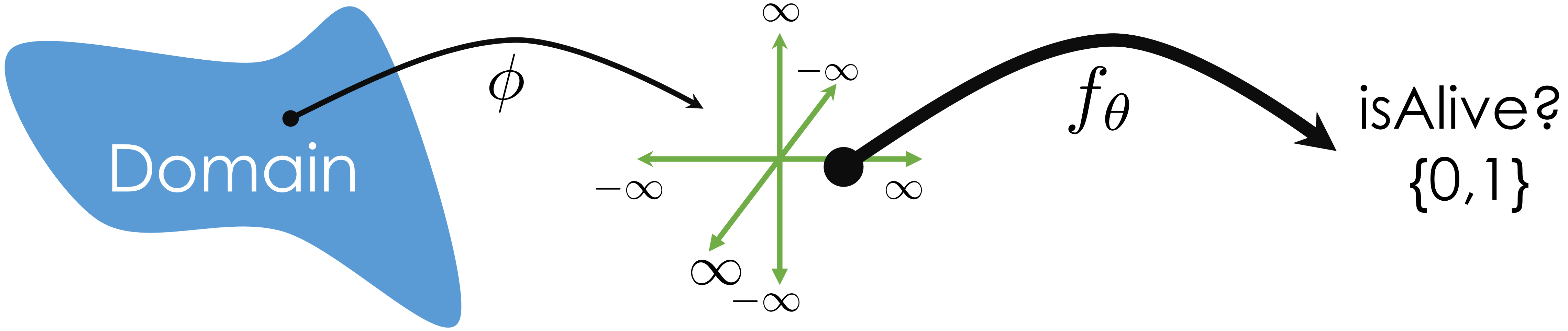


Squared Loss

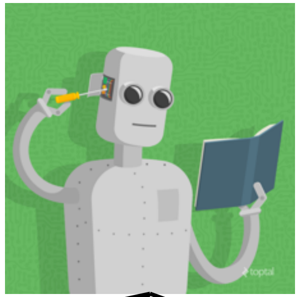
$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2 + \lambda \mathbf{R}(\theta)$$

Regularization

Classification



Taxonomy of Machine Learning



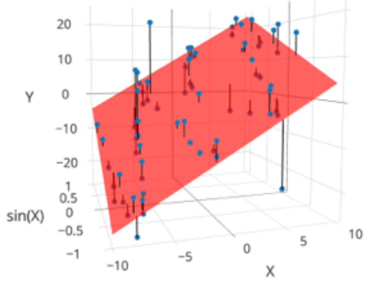
Labeled Data → **Supervised Learning**

Reward → **Reinforcement Learning (covered later)**

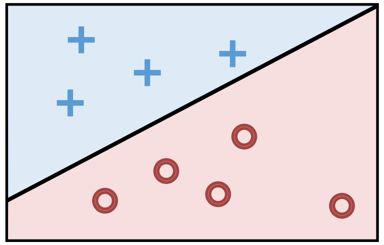
Unlabeled Data → **Unsupervised Learning**

Supervised Learning

Quantitative Response → Regression



Categorical Response → Classification

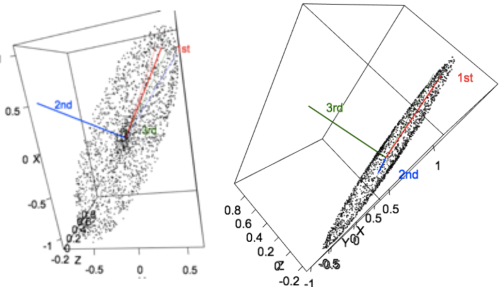


Reinforcement Learning (covered later)

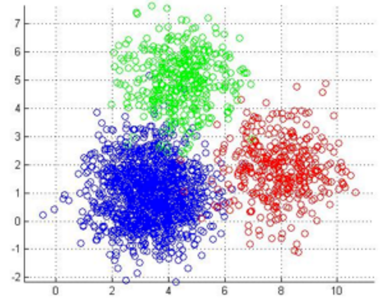
Alpha Go

Unsupervised Learning

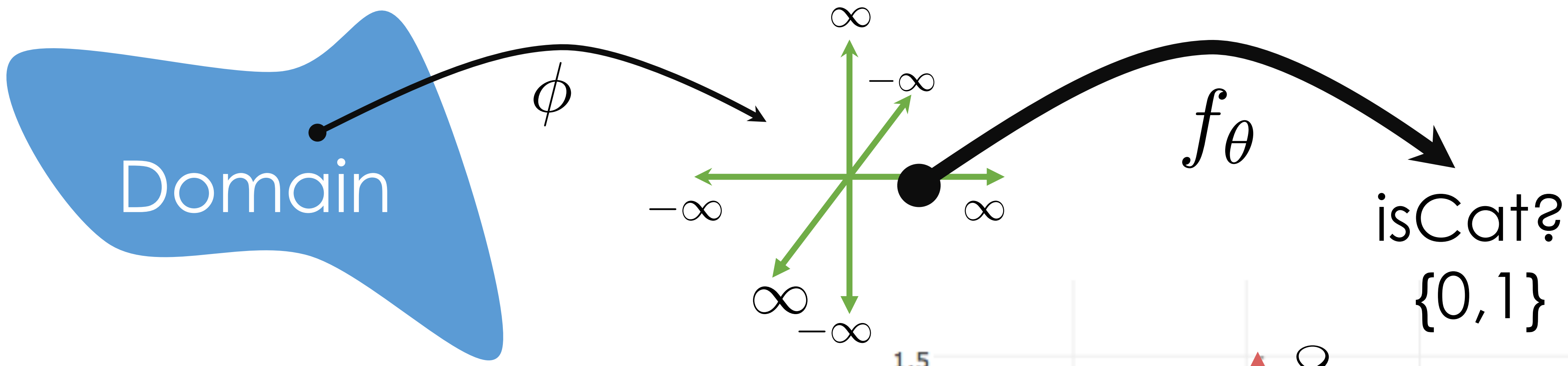
Dimensionality Reduction

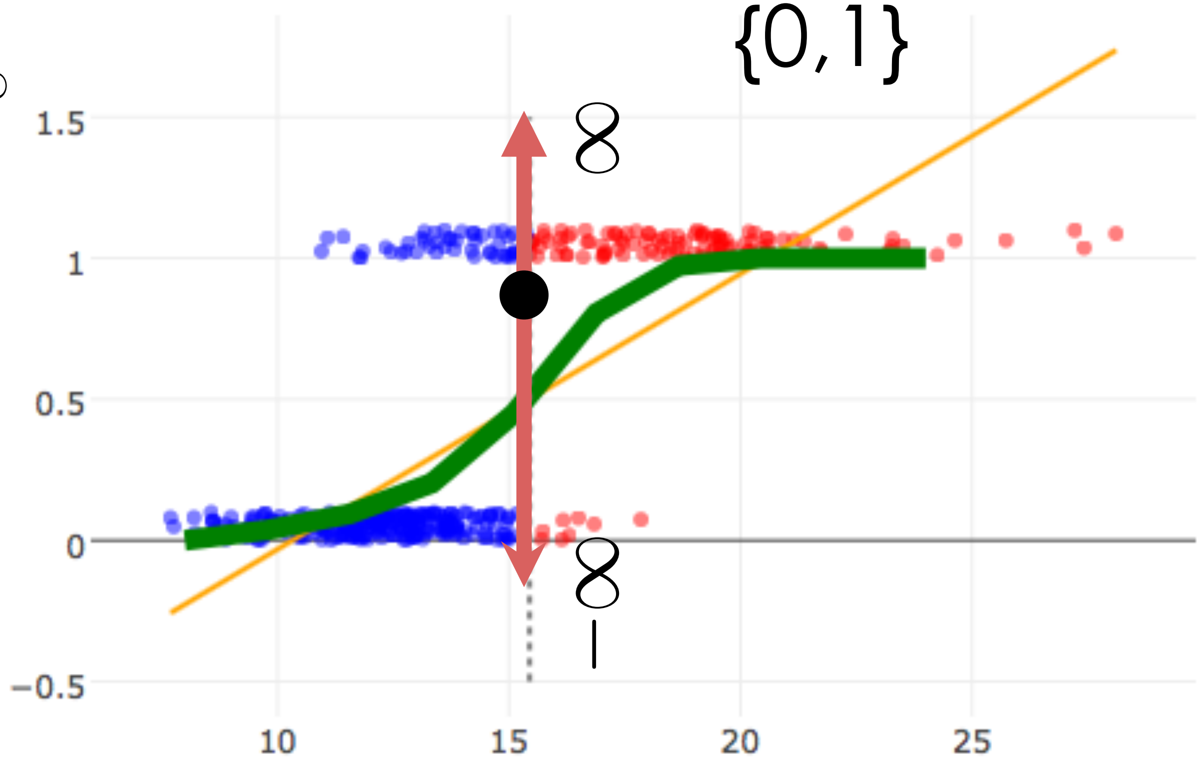


Clustering



Classification





Can we just use least squares?

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - f_\theta(x_i)\right)^2 + \lambda \mathbf{R}(\theta)$$

Defining the Loss

Could we use the Squared Loss

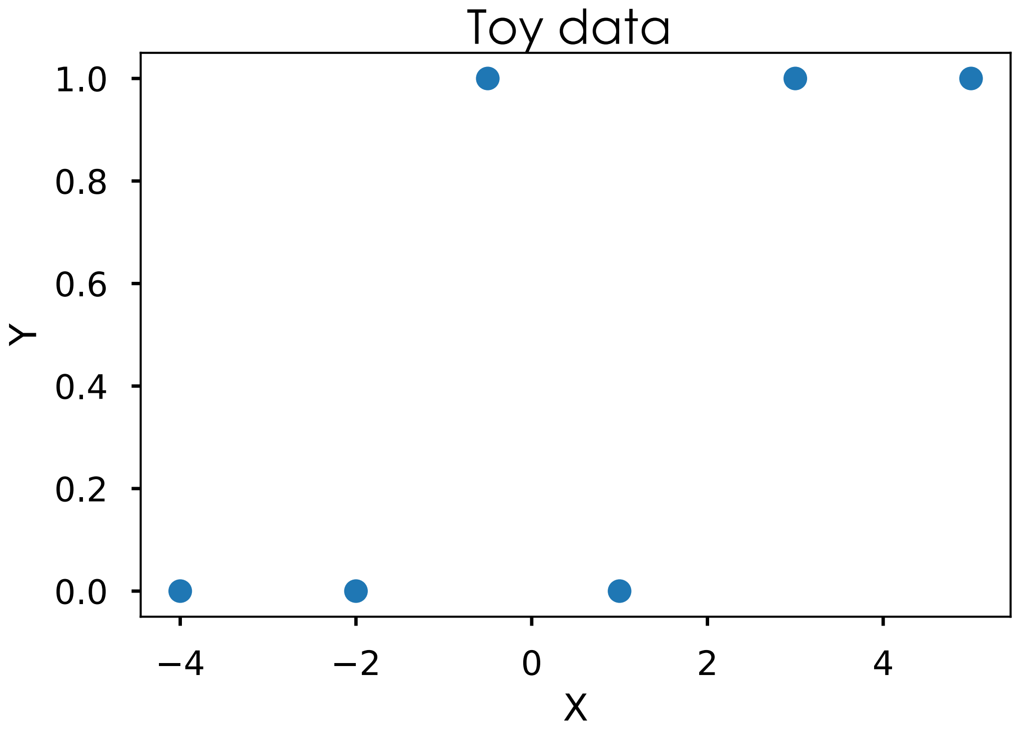
- What about squared loss and the new model:

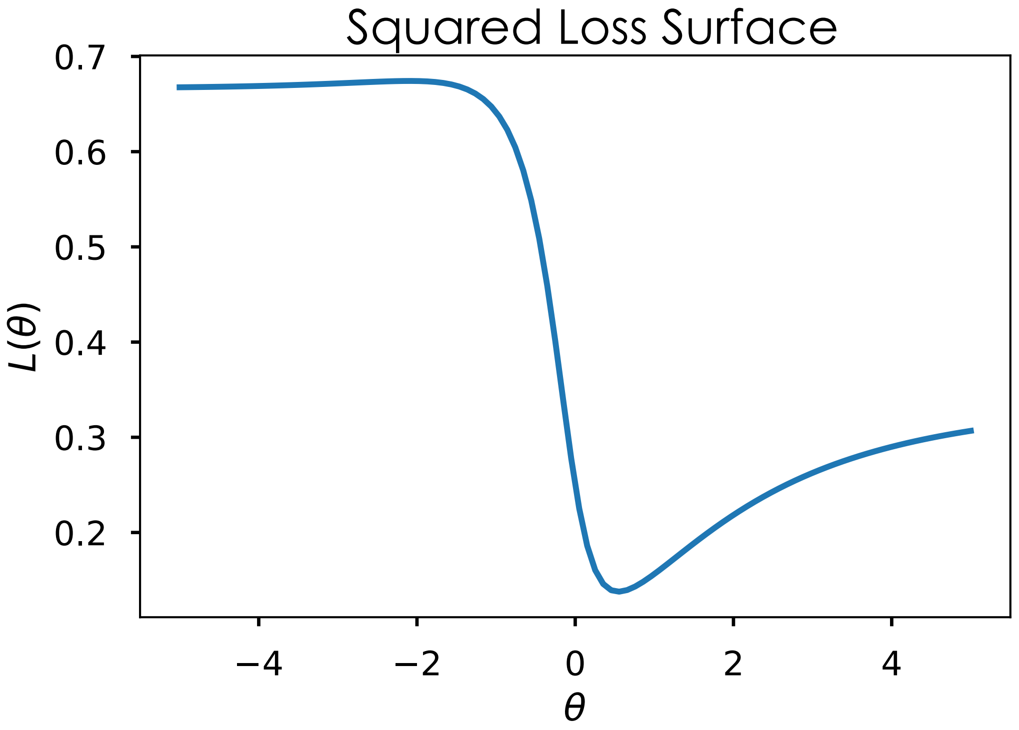
$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sigma\left(\phi(x_i)^T \theta \right) \right)^2$$

  - Tries to match probability with 0/1 labels.
  - Occasionally used in some neural network applications
  - **Non-convex!**

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sigma\left(\phi(x_i)^T \theta\right)\right)^2$$

- Tries to match probability with 0/1 labels.
- Occasionally used in some neural network applications
- **Non-convex!**





Defining the Cross Entropy Loss

Loss Function

- We want our model to be close to the data:

$$\hat{\mathbf{P}}_\theta\left(y = 1 \,|\, x\right) \;\approx\; \mathbf{P}\left(y = 1 \,|\, x\right)$$

- Example: (cute or not)?

x = 

y = 1 "cute"

	Cute	Not Cute
Observed Probability	$\mathbf{P}\left(y = 1 \,\|\, x\right)$ = 1.0	$\mathbf{P}\left(y = 0 \,\|\, x\right)$ = 0.0
Predicted Probability	$\hat{\mathbf{P}}_\theta\left(y = 1 \,\|\, x\right)$ = 0.8	$\hat{\mathbf{P}}_\theta\left(y = 0 \,\|\, x\right)$ = 0.2

The Loss for Logistic Regression

- Average **cross entropy** (simplified):

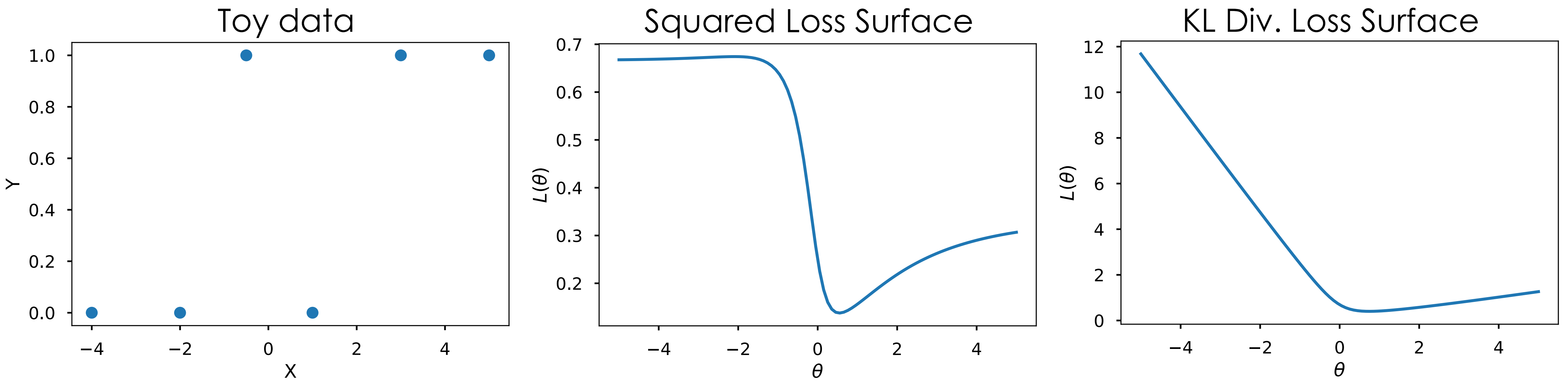
$$\arg\min_{\theta} -\frac{1}{n}\sum_{i=1}^{n}\left(y_i\phi(x_i)^T\theta + \log\left(\sigma\left(-\phi(x_i)^T\theta\right)\right)\right)$$

  - Equivalent to (derived from) **minimizing the KL divergence**
  - Also equivalent to **maximizing the log-likelihood of the data** … (not covered in Data100 this semester)

Is this loss function reasonable?

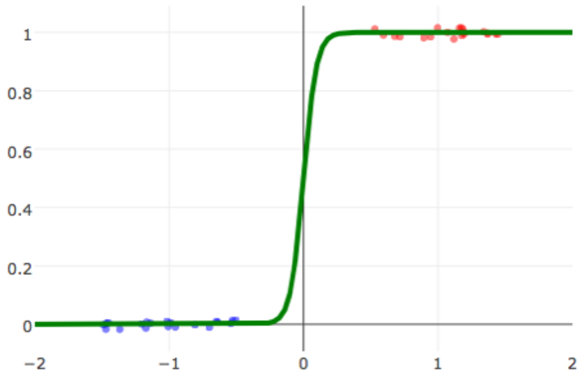
Convexity Using Pictures

$$\arg\min_{\theta} -\frac{1}{n}\sum_{i=1}^{n}\left(y_i\phi(x_i)^T\theta + \log\left(\sigma\left(-\phi(x_i)^T\theta\right)\right)\right)$$



Linearly Separable Data

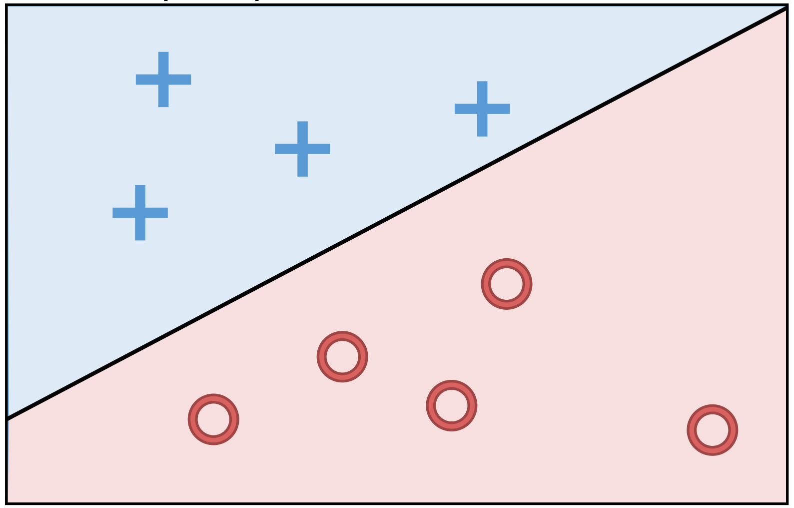
- A classification dataset is said to be linearly separable if there exists a hyperplane that separates the two classes.
- If data is linearly separable, logistic regression requires regularization



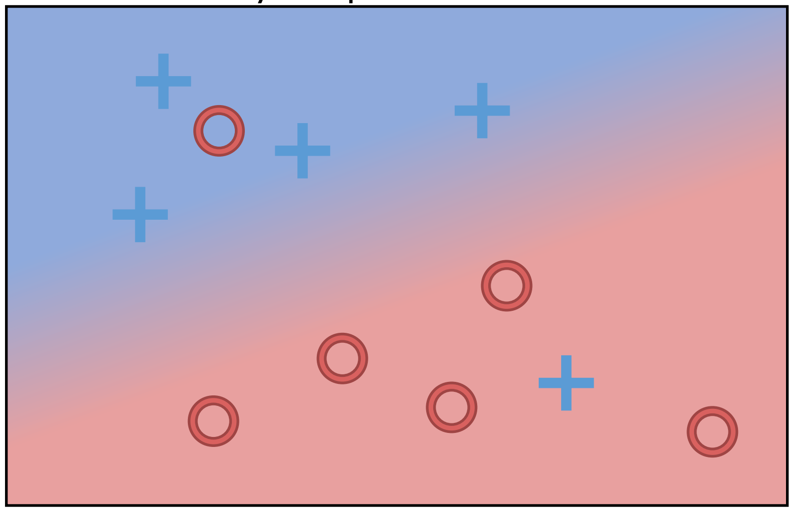
Weights go to infinity!

Solution?

Linearly Separable Data



Not Linearly Separable Data



Adding Regularization to Logistic Regression

$$\arg\min_{\theta} -\frac{1}{n}\sum_{i=1}^{n}\left(y_i \phi(x_i)^T\theta + \log\left(\sigma\left(-\phi(x_i)^T\theta\right)\right)\right) + \lambda\sum_{j=1}^{d}\theta_j^2$$

- Prevents weights from diverging on linearly separable data

Earlier Example