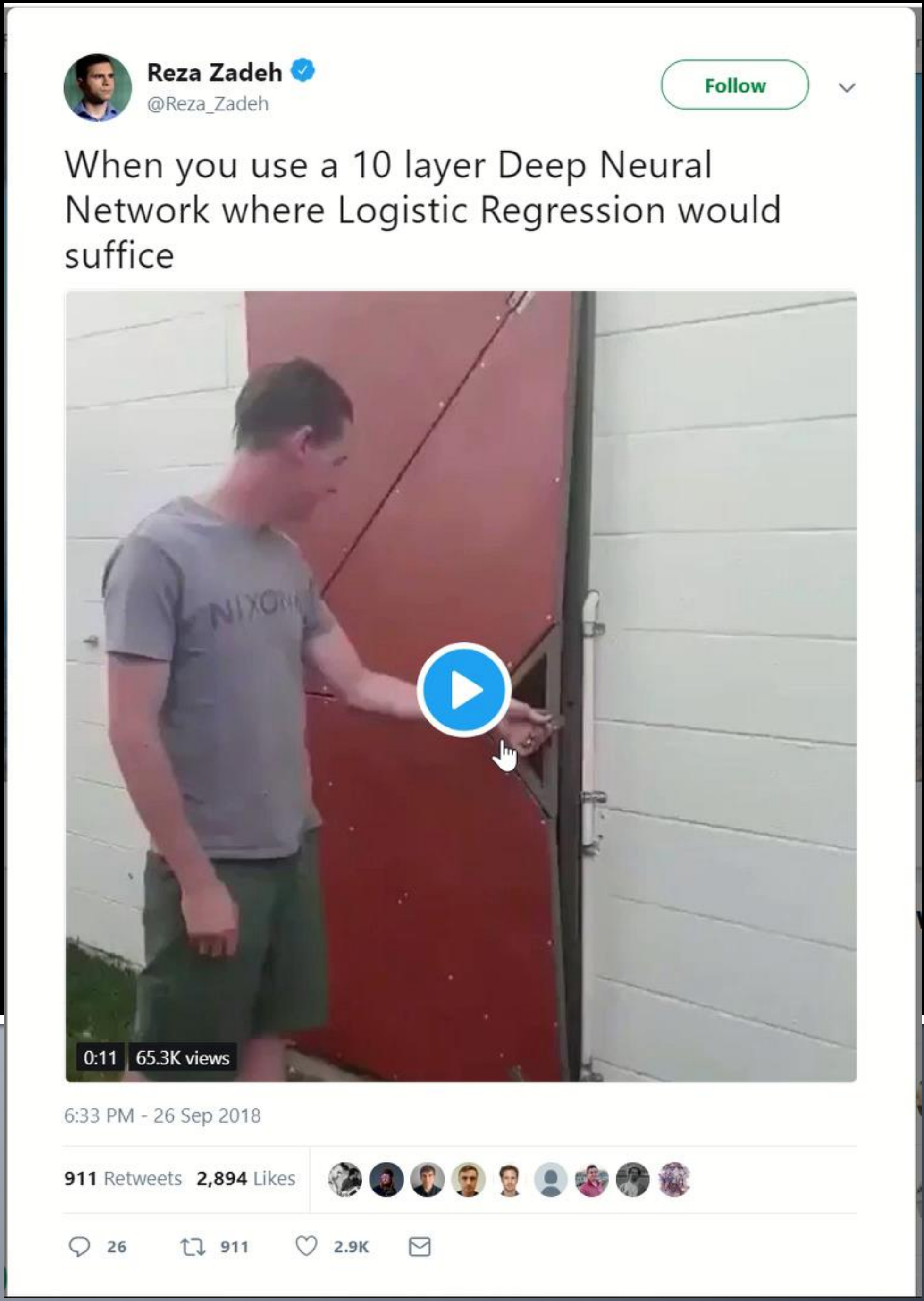
INFO 251: Applied Machine Learning

Logistic Regression



Announcements

- Assignment 4 will be posted today/tomorrow

Key Concepts (last lecture)

- Overfitting
- Regularization: Intuition
- Regularization: Cost function adjustment
- Ridge
- Lasso
- Cross-validation of regularization hyperparameters
- Coefficient plots
- Logistic regression
- Sigmoid function
- Odds ratios

Course Outline

- Causal Inference and Research Design
  - Experimental methods
  - Non-experiment methods
- **Machine Learning**
  - Design of Machine Learning Experiments
  - **Linear Models and Gradient Descent**
  - Non-linear models
  - Fairness and Bias in ML
  - Neural models
  - Deep Learning
  - Practicalities
  - Unsupervised Learning
- Special topics

Outline

- Logistic regression (interpretation)
- Logistic regression (prediction and gradient descent)
- Support vector machines
- Kernels

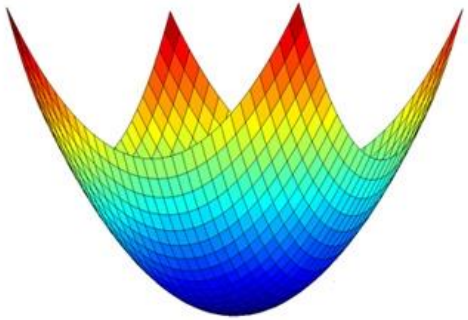
Outline

- Logistic regression (inference)
- **Logistic regression (prediction & gradient descent)**
- Support vector machines
- Kernels

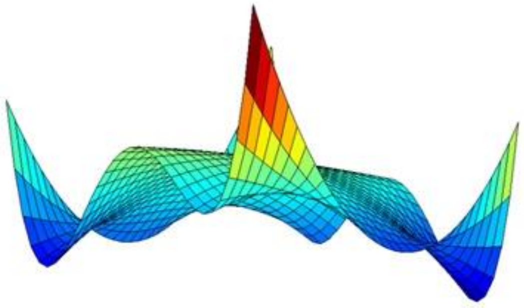
Cost functions and convexity

- How to know if cost function is convex?
- Intuition: Need that "bowl" shape
- Formally
  - A function $J(\theta)$ is convex if its Hessian (2nd order derivative) is positive semi-definite: $H = \nabla^2 \mathrm{J}(\theta) \geq 0$
    - (All eigenvalues are non-negative)
  - In practice, computing Hessian can be difficult, and only works if $J(\theta)$ is twice differentiable

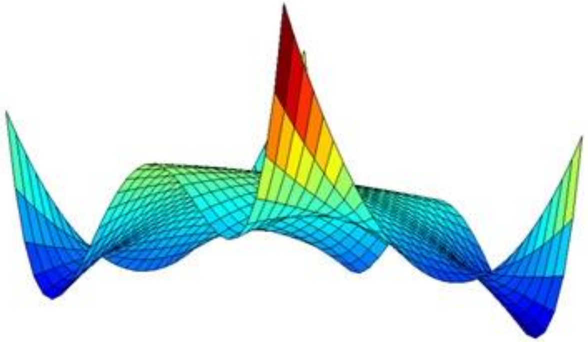
Convex ☺



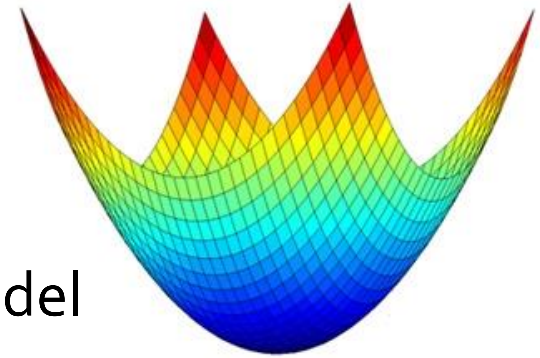
Non-convex ☹



Logistic Regression: Cost function

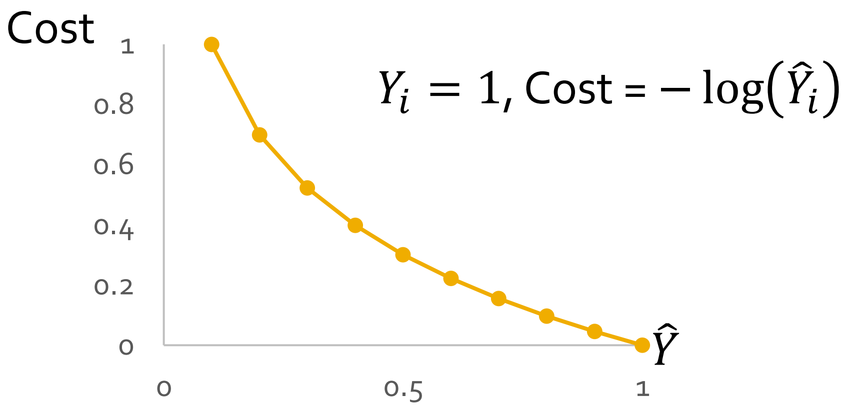


- Cost Functions:
  - Linear regression: $J(\alpha, \beta) = \frac{1}{2N} \sum_{i=1}^{N} (Y_i - \alpha - \beta X_i)^2$
  - Why not $J(\alpha, \beta) = \frac{1}{2N} \sum_{i=1}^{N} \left(Y_i - \frac{1}{1+e^{-(\alpha+\beta X_i)}} \right)^2$
- Not convex ☹
  - Sigmoid function is complex
  - When sigmoid is combined with Squared Error Loss, $J(\alpha, \beta)$ not convex…
  - Susceptible to local minima
- Instead, we use something different
  - (derived from negative log-likelihood of Bernoulli probability model

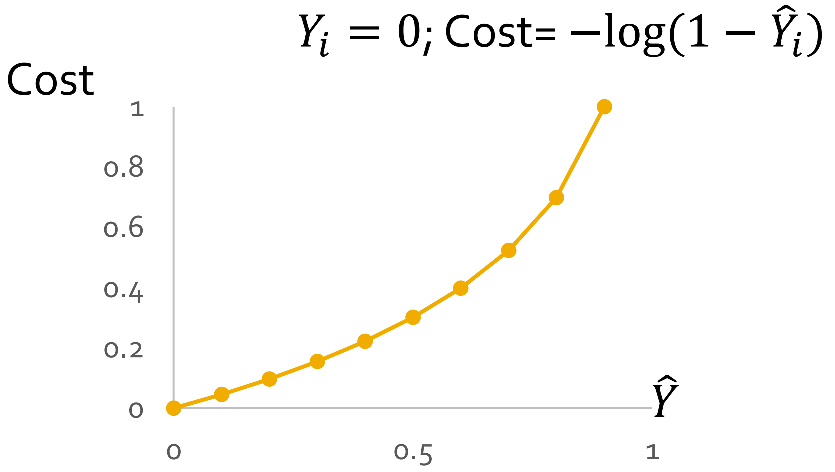


Logistic Regression: Cost function

- Cost Function (think of $\hat{Y}_i = \frac{1}{1+e^{-(\alpha+\beta X_i)}}$)
  - $\text{Cost}(\hat{Y}_i, Y_i) = \begin{cases} -\log(\hat{Y}_i) & \text{if } Y_i = 1 \\ -\log(1-\hat{Y}_i) & \text{if } Y_i = 0 \end{cases}$
  - $\text{Cost}(\hat{Y}_i, Y_i) = -Y_i \cdot \log(\hat{Y}_i) - (1-Y_i) \cdot \log(1-\hat{Y}_i)$



- This is convex:
  - If $Y_i = 1$, what is cost if $\hat{Y}_i = 1$? What if $\hat{Y}_i = 0$?
    - No cost if model predicts 1
    - Penalizes mistakes
  - If $Y_i = 0$, what is cost if $\hat{Y}_i = 1$? if $\hat{Y}_i = 0$?
    - No cost if model predicts 0
    - Penalizes mistakes



Logistic Regression: Gradient Descent

- Given the cost function $J(\theta)$, we now want to minimize:
  - $J(\theta) = -\frac{1}{N}\sum_{i=1}^{N} Y_i \cdot \log \hat{Y}_i + (1 - Y_i)\log\left(1 - \hat{Y}_i\right)$
- Gradient Descent!
  - $\theta \leftarrow \theta - R\frac{\partial}{\partial\theta}J(\theta)$
- With revised cost function, $\frac{\partial}{\partial\theta}J(\theta) = -\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)X_i$
  - Note similarities to linear regression! But not identical:
  - Logistic regression: $\hat{Y}_i = \frac{1}{1+e^{-(\alpha+\beta X_i)}}$
- Gradient Descent Algorithm (logistic regression)
  - Repeat until convergence:
  - $\beta \leftarrow \beta + R\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)X_i$
  - in other words: $\beta \leftarrow \beta + R\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \frac{1}{1+e^{-(\alpha+\beta X_i)}}\right) X_i$

Outline

- Logistic regression (inference)
- Logistic regression (prediction and gradient descent)
- **Support vector machines**
- Kernels

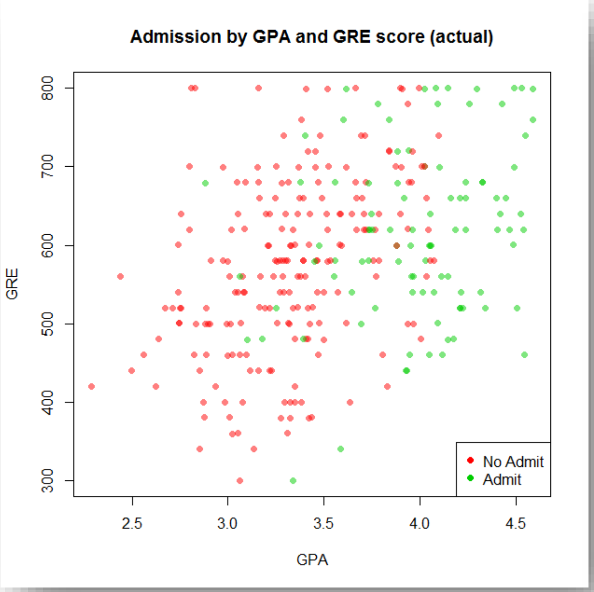
Logistic Regression: Linear decision boundary

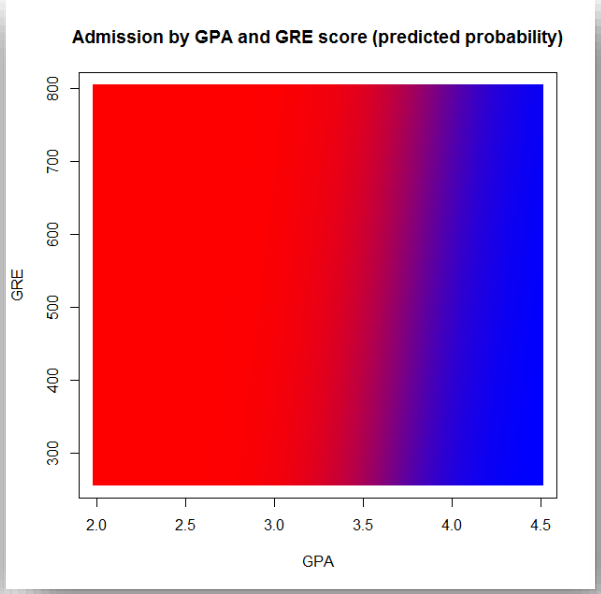
- Logistic regression is one (very) common binary classifier
  - Prediction $\hat{Y}_i$ can be interpreted as probability that $Y_i = 1$
  - To then make a binary prediction, a threshold is applied
    - (typically, at 0.50)
    - (AUC provides a "threshold-agnostic" measure of performance)

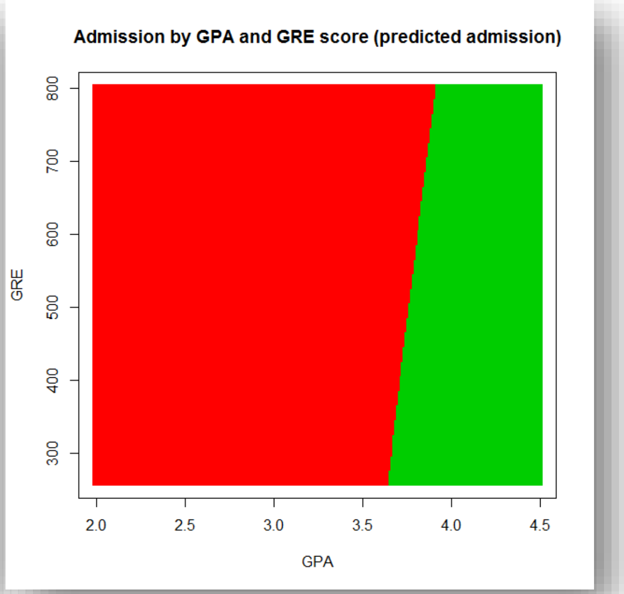
- This creates a linear decision boundary
  - i.e., the decision boundary can be expressed as a linear function (a "hyperplane")

Logistic Regression: Linear decision boundary

- Example: admission vs. GRE and GPA
  1. Start with raw data
  2. Fit logistic regression
  3. Threshold converts predicted probabilities to classifications

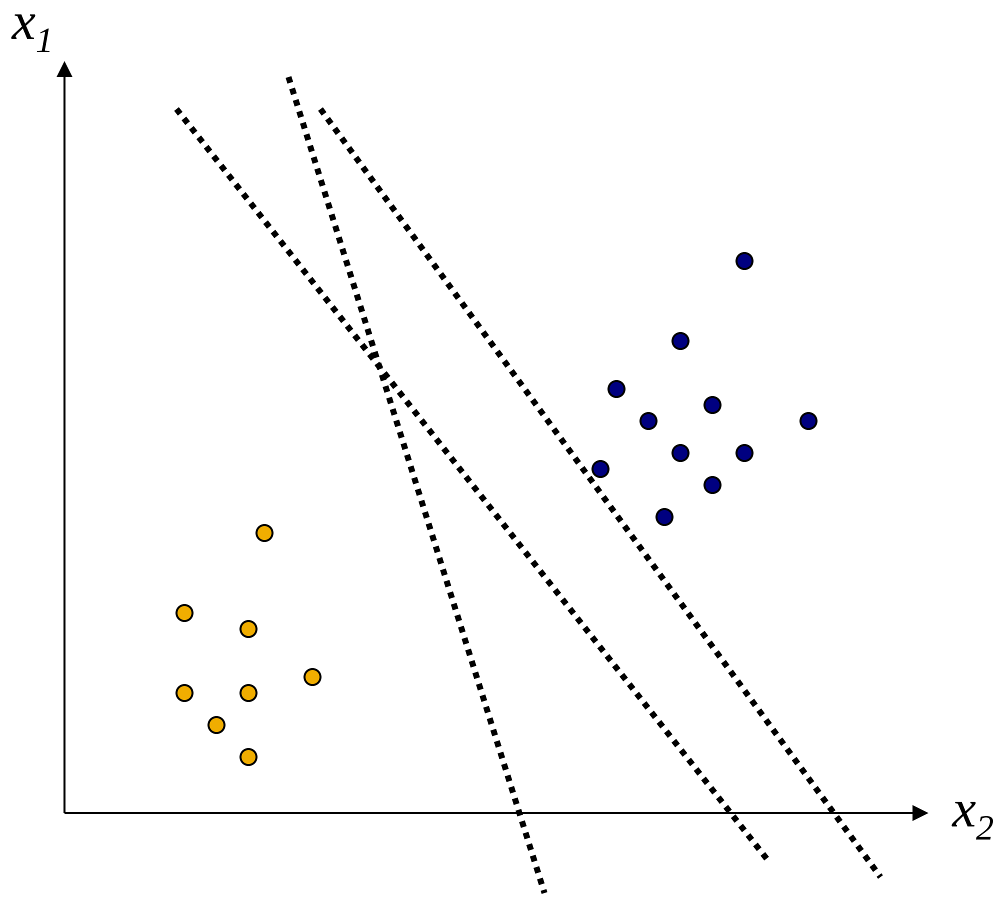






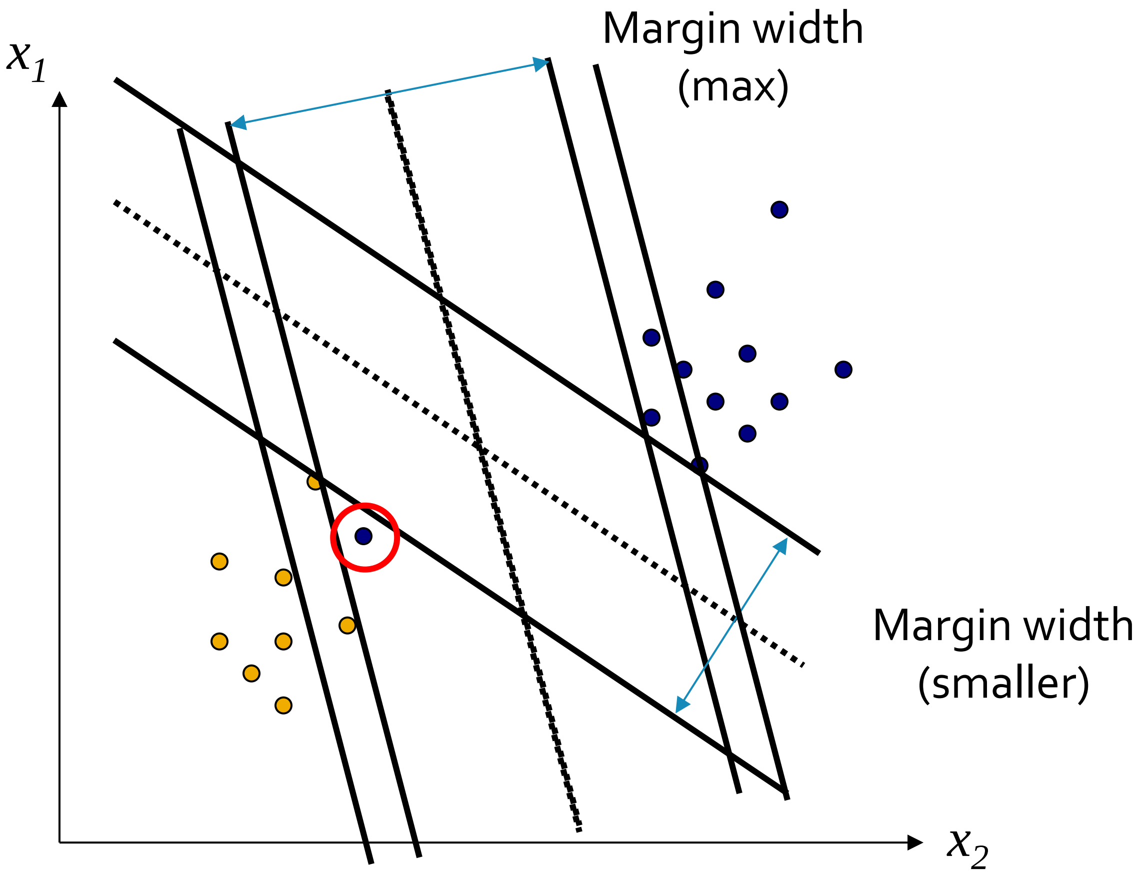
Support Vector Classifiers: Intuition

- Often there are multiple possible decision boundaries that perform equivalently on the training data



Support Vector Classifiers: Intuition

- Idea: Select the hyperplane that maximizes the "margin"
  - "Margin": shortest distance between training observations and threshold
  - Example of "max margin classifier"
- Note: max margin is brittle!
  - For this reason, typically want to use a "soft margin classifier"
  - Allows misclassifications w/in margin
  - Use cross-val to determine margin width



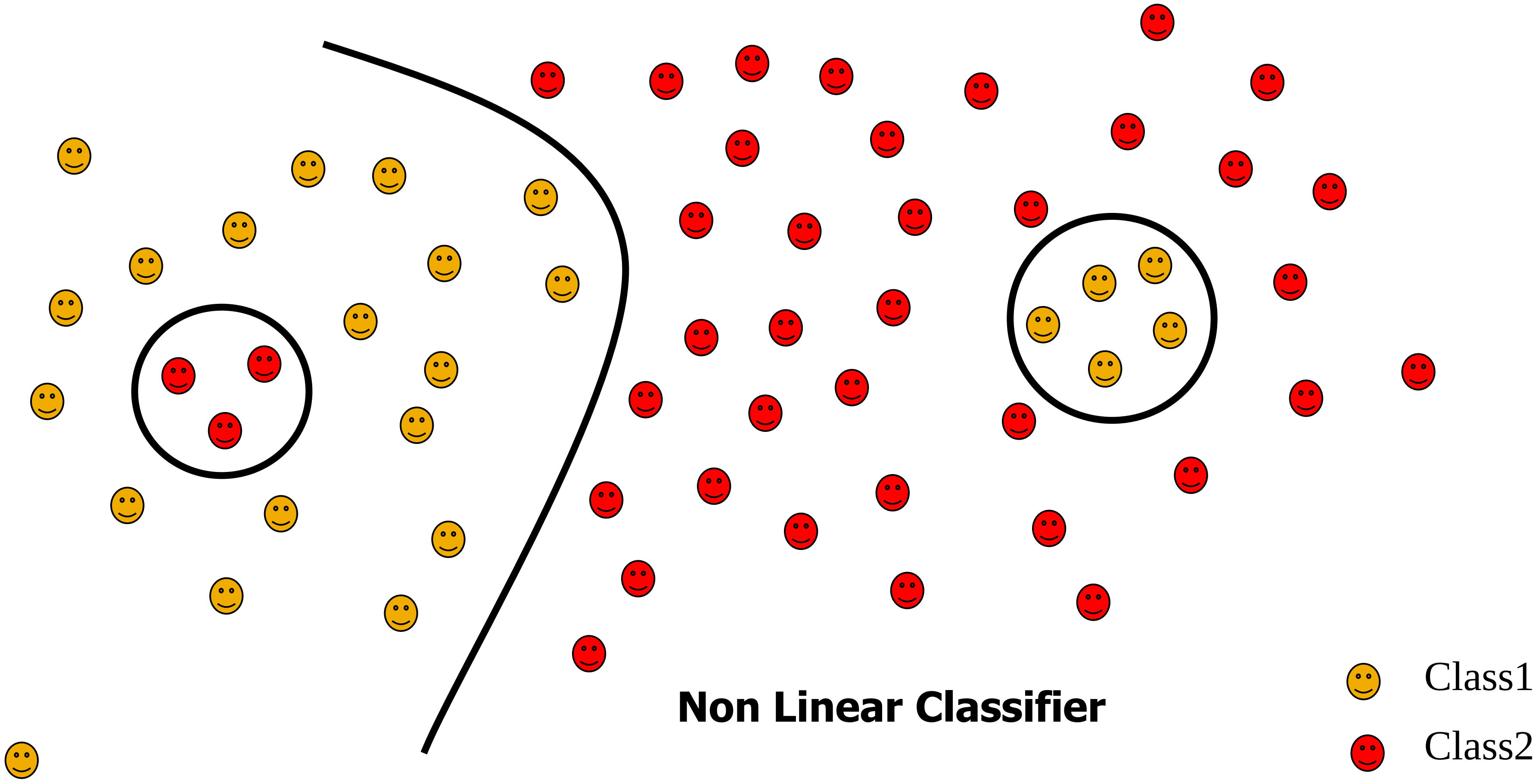
Linear models: Recap

- Linear models rely on some notion of a linear boundary (i.e., a hyperplane)
- But real-world data are typically not linearly separable
- Some classifiers just make a decision as to which class an object is in; others estimate class probabilities

Outline

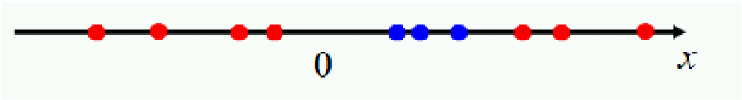
- Logistic regression (inference)
- Logistic regression (prediction and gradient descent)
- Support vector machines
- **Kernels**

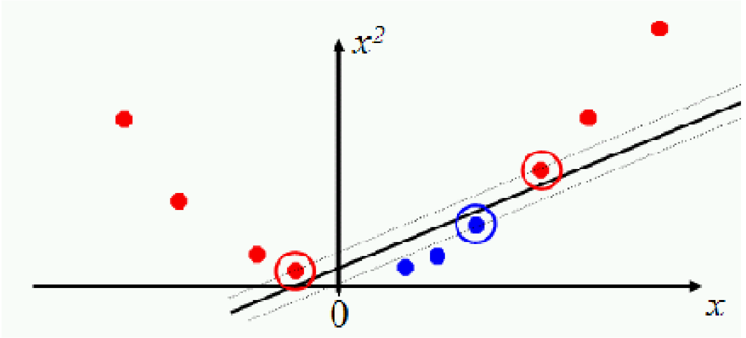
Nonlinearly separable data



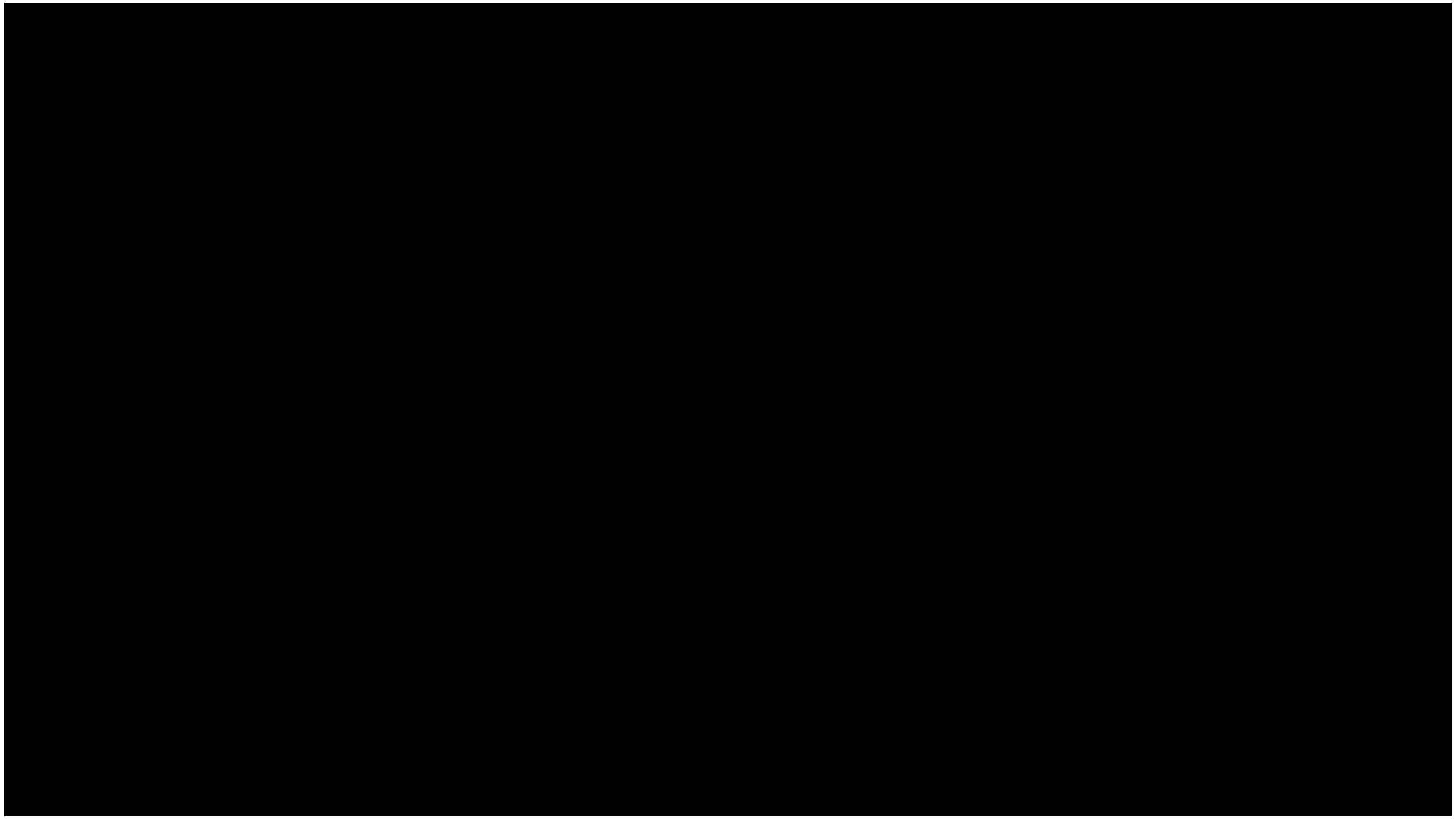
Extending linear models

- We are modeling y with feature x
  - Classes are not separable with this feature
- One solution: non-linear classifier
  - E.g., k-NN
- Another solution: use kernels!
  - Transforms data
  - E.g., x^2





Kernel Visualization



Support Vector Machines (SVM)

- SVM: A general-purpose support vector classifier
  - Combines kernel functions (basis functions) w/ support vector classifiers
  - Common kernels: polynomial kernel, radial basis function (RBF)
- Main idea
  - Kernel is used to project data into higher-dimensional space
  - Support vector classifier finds best soft-margin classifier
  - Cross-validation can be used to tune kernel
  - Other bells and whistles for regularization, efficiency (see ESL 12.3)

Key Concepts (this lecture)

- Sigmoid cost function
- Gradient descent with logistic regression
- Odds ratios
- Support vector machines
- Hard vs. soft margins
- Kernel functions

Linear Models: Example Quiz Question

- True or False: If the cost function is continuous and differentiable, and the learning rate is sufficiently small, gradient descent will eventually converge to the global minimum.

For Next Class:

- Read:
  - Chapters 5 and 6 of Daume