Day 6 - Linear Regression and Logistic Regression

Agenda:

- Linear Regression
  - Examples
  - Issues to Pay Attention To with Linear Regression
- Classification and Logistic Regression
  - Training classifiers
  - Evaluating classifiers

More thoughts on square capital example and whether to approach problem as regression or classification

CS 6140: Machine Learning — Fall 2021— Paul Hand

HW 2
Due: Wednesday September 29, 2021 at 2:30 PM Eastern time via Gradescope.

Names: [Put Your Name(s) Here]

You can submit this homework either by yourself or in a group of 2. You may consult any and all resources. You may submit your answers to this homework by directly editing this tex file (available on the course website) or by submitting a PDF of a Jupyter or Colab notebook. When you upload your solutions to Gradescope, make sure to tag each problem with the correct page.

Question 1. *In this problem, you will fit polynomials to one-dimensional data using linear regression.*

Assemble Data

(a) Generate training data (x_i, y_i) for $i = 1 \dots 8$ by $x_i \sim \text{Uniform}([0,1])$, and $y_i = f(x_i) + \varepsilon_i$, where $f(x) = 1 + 2x - 2x^2$ and $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\sigma = 0.1$. Plot the training data and the function f.

Response:

Train a ML Model Learn θ

(b) In this problem, you will find the best fit degree d polynomial for the above data for each d between 0 and 7. Find it with least squares linear regression by minimizing the training mean squared error (MSE)

$$\min_\theta \frac{1}{2} \sum_{i=1}^{8} \left(y_i - \sum_{k=0}^{d} \theta_k x_i^k \right)^2 \tag{1}$$

using the Normal Equations. Use numpy.linalg.solve to solve the Normal Equations instead of computing a matrix inverse. On 8 separate plots, plot the data and the best fit degree-d polynomial.

Response:

Evaluate Model (use θ you already learned)

(c) Plot the MSE with respect to the training data (training MSE) as a function of d. Which value of d provided the lowest training MSE?

Response:

(d) Generate a test set of 1000 data points sampled according to the same process as in part (a). Plot the MSE with respect to the test data (test MSE) as a function of d. Which value of d provided the lowest test MSE?

Response:

Question 2. *Linear regression using gradient descent and TensorFlow*

Least Squares Formulation for Linear Regression (for a general model)

Given: $D = \{(X^{(i)}, y_i)\}_{i=1\cdots n}$, $X^{(i)} \in \mathbb{R}^d$, $y_i \in \mathbb{R}$

Model: $y = \underbrace{\theta_1 g_1(X^{(i)}) + \cdots \theta_k g_k(X^{(i)})}_{f_\theta(x)} + \text{Error}$

Want $y \approx \mathbb{X}\theta$

$n \times k$

w/ $\mathbb{X} = \begin{pmatrix} g_1(X^{(1)}) & g_2(X^{(1)}) \cdots & g_k(X^{(1)}) \\ \vdots & \ddots & \vdots \\ g_1(X^{(n)}) & \cdots & g_k(X^{(n)}) \end{pmatrix} = \begin{pmatrix} - g(X^{(1)}) - \\ - g(X^{(2)}) - \\ \vdots \\ - g(X^{(n)}) - \end{pmatrix}$

Find

$$\min_\theta \frac{1}{2}\|y - \mathbb{X}\theta\|^2$$

— minimizes sum of squares of errors

where $y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$, $\theta = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_k \end{pmatrix}$

$e_i = y_i - \hat{y}_\theta(X^{(i)})$

Solution given by <u>Normal Equations</u>

$$X^t X \theta = X^t y \Rightarrow \theta = (X^t X)^{-1} X^t y.$$

Examples of setting up and solving linear regression

Find best fit cubic through 1d data

Data $= \{(x_i, y_i)\}_{i=1\cdots n}$ w/ $x_i, y_i \in \mathbb{R}$

Model $y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \text{noise}$

$$\begin{pmatrix} 1 & x & x^2 & x^3 \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix}$$

Find

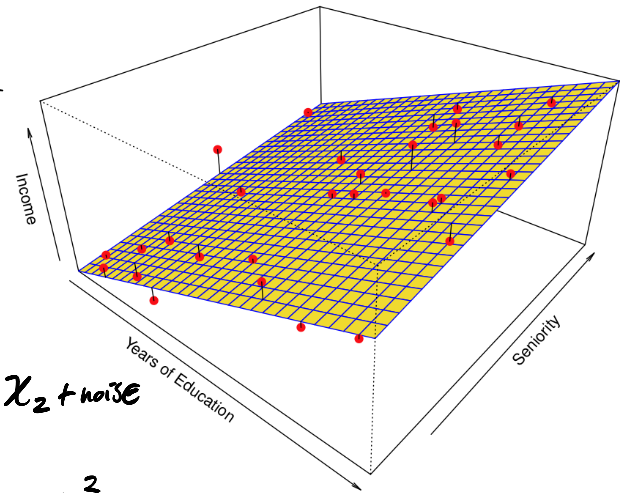
$$\min_\theta \frac{1}{2} \| y - \mathbb{X}\theta \|^2$$

where $y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$, $\theta = \begin{pmatrix} \theta_0 \\ \vdots \\ \theta_3 \end{pmatrix}$, $\mathbb{X} = \begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 \end{pmatrix}$

Solution given by Normal Equations

$$X^t X \theta = X^t y \quad \Rightarrow \quad \theta = (X^t X)^{-1} X^t y.$$

Find best fit plane



Data $= \{(x^{(i)}, y_i)\}_{i=1\cdots n}$

$x^{(i)} \in \mathbb{R}^2$, $y_i \in \mathbb{R}$

Model: $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \text{noise}$

Find

$$\min_{\theta} \frac{1}{2} \| y - X\theta \|^2$$

where $y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$, $\theta = \begin{pmatrix} \theta_0 \\ \vdots \\ \theta_3 \end{pmatrix}$, $X = \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} \\ & \vdots & \\ 1 & x_1^{(n)} & x_2^{(n)} \end{pmatrix}$

$$y = \begin{pmatrix} 1 & x_1 & x_2 \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{pmatrix}$$

Solving and Optimization Problem using Gradient Descent

$$\min_{x} \; f(x)$$

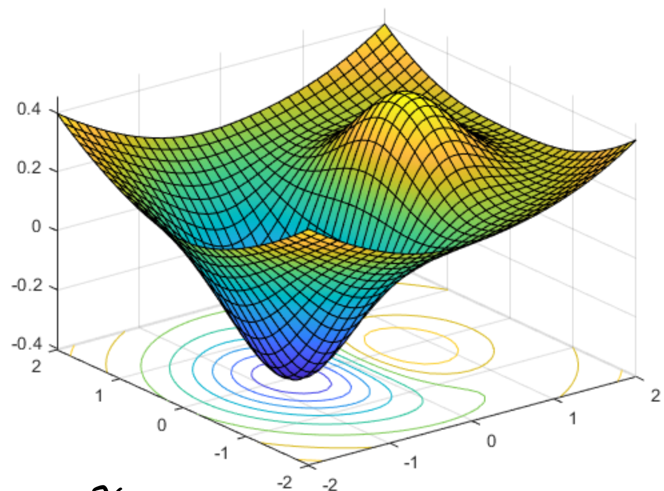
Gradient descent: Take successive steps "downhill"

$$x^{i+1} = x^{i} - \alpha \nabla f(x^{i})$$

α: step size, learning rate

$-\nabla f$ points in direction of steepest descent

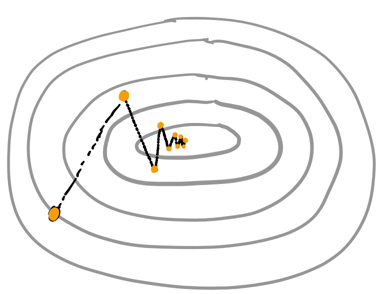
$f(x_1, x_2)$



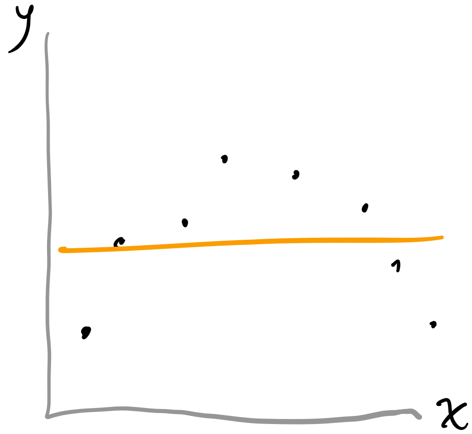
x_2 x_1

level sets

Depiction of gradient descent



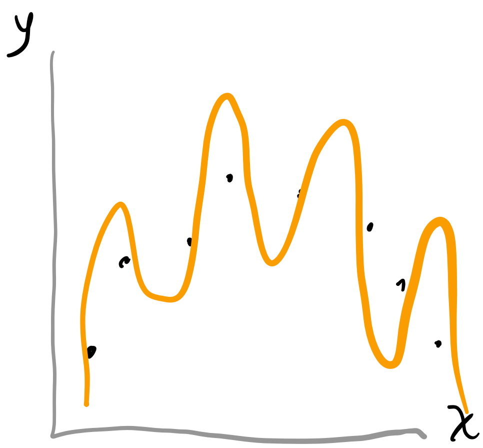
Things that can go wrong: Underfitting and Overfitting



Underfitting

model $y = \theta_0$

model not expressive enough



Overfitting

model is too expressive

it can fit noise

Things that can go wrong: numerical instability

$$X^t X \theta = X^t y$$

Other topics:

What happens when there is fewer data than features?

$$y \approx \underset{n \times k}{X} \underset{k}{\theta}$$

y: n

If $n \geq k$, usually $X^t X$ is invertible

There is a unique θ.

If $n < k$, mly many solutions

(X has a null space)

What happens if there are outliers in the data?

$$\min_\theta \frac{1}{2} \| y - X\theta \|^2$$

1 outlier can show estimate of Θ arbitrarily much

Issue w/ MSE

more robust & median

$$\min_\theta \sum_{i=1}^{n} \| y - X\theta \|_1$$

How do you deal with categorical features?

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$$x_2 = \begin{cases} 1 & \text{if person } i \text{ is consultant} \\ 0 & \text{o'wise} \end{cases}$$

indicator

Be careful about whether you want to view your problem as a prediction task

Classification and Logistic Regression

Viewing Regression and Classification as function estimation problems

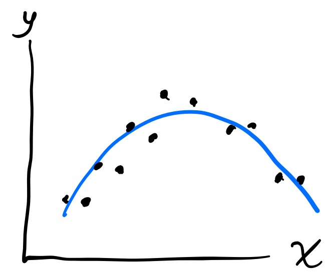
Regression: predict a continuous value

Let $f: \mathbb{R}^d \to \mathbb{R}$

$y = f(x) + \text{noise}$

Given: $\{(x^{(i)}, y_i)\}_{i=1\dots n}$

Find: f



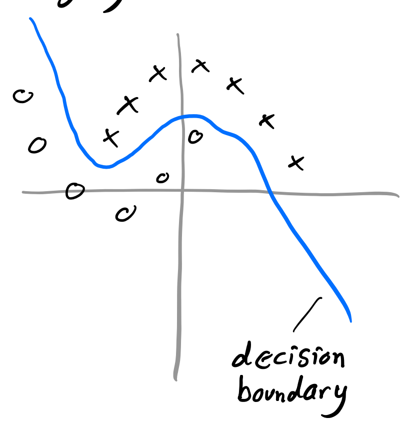
Classification: predict membership in a category

Let $f: \mathbb{R}^d \to \left\{\begin{matrix} \text{cat } 1 \\ \vdots \\ \text{cat } m \end{matrix}\right\}$

$y = f(x) + \text{noise}$

Given: $\{(x^{(i)}, y_i)\}_{i=1\dots n}$

Find: f



Terminology:

x – input variables, predictors, independent vars, features

y – response, dependent variable, output variable

f – model, predictor, hypothesis

Parametric Approach: Choose a model for f with unknown parameters. Estimate the parameters.

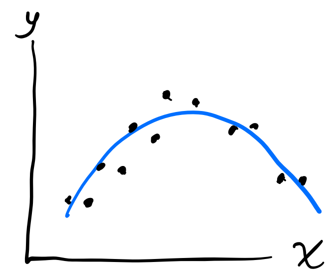
Parametric Regression: predict a continuous value

Model $f_\theta : \mathbb{R}^d \to \mathbb{R}$

$y = f_\theta(x) + \text{noise}$

Given: $\{(x^{(i)}, y_i)\}_{i=1\ldots n}$

Find: θ



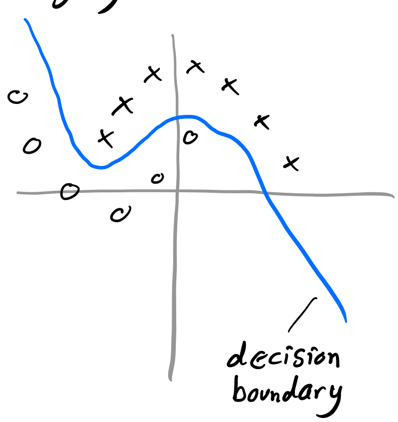
Parametric Classification: predict membership in a category

Let $f_\theta : \mathbb{R}^d \to \begin{Bmatrix} \text{cat } 1 \\ \vdots \\ \text{cat } m \end{Bmatrix}$

$y = f_\theta(x) + \text{noise}$

Given: $\{(x^{(i)}, y_i)\}_{i=1\ldots n}$

Find: θ



Approach for estimating θ:

Select a model for f w/ parameters θ

&

minimize the loss between training labels and predictions on training data

$$\min_{\theta} \sum_{i=1}^{n} L(y_i, f_\theta(x^{(i)}))$$

loss function

prediction of y

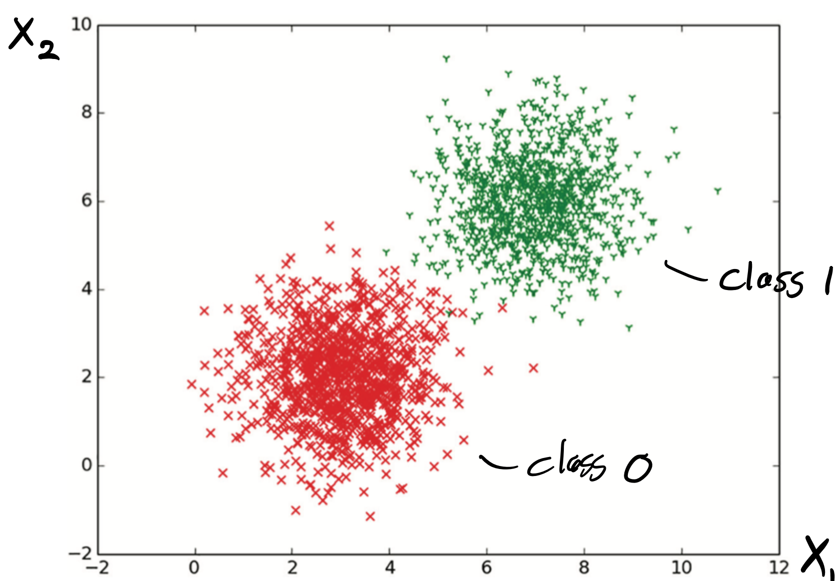
Example: linear regression $L(y, \hat{y}) = |y - \hat{y}|^2$

"square loss" or ℓ^2 loss

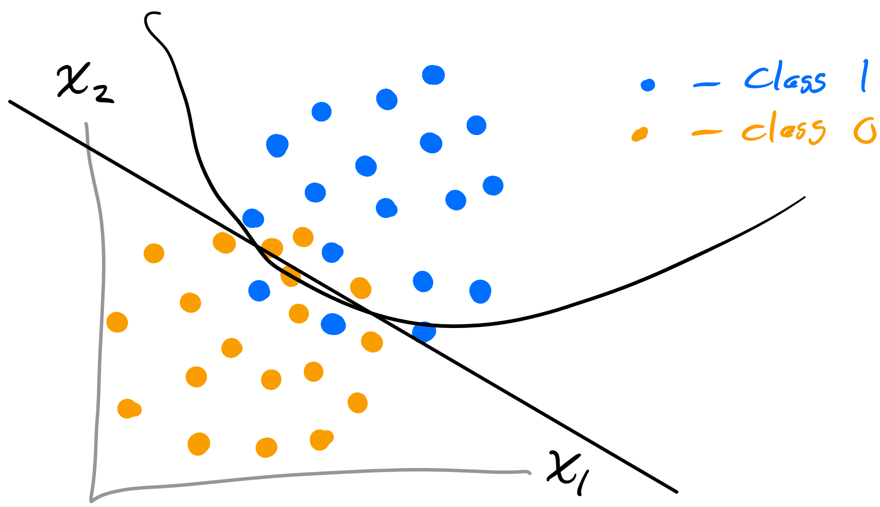
Binary Classification in 2D with logistic regression

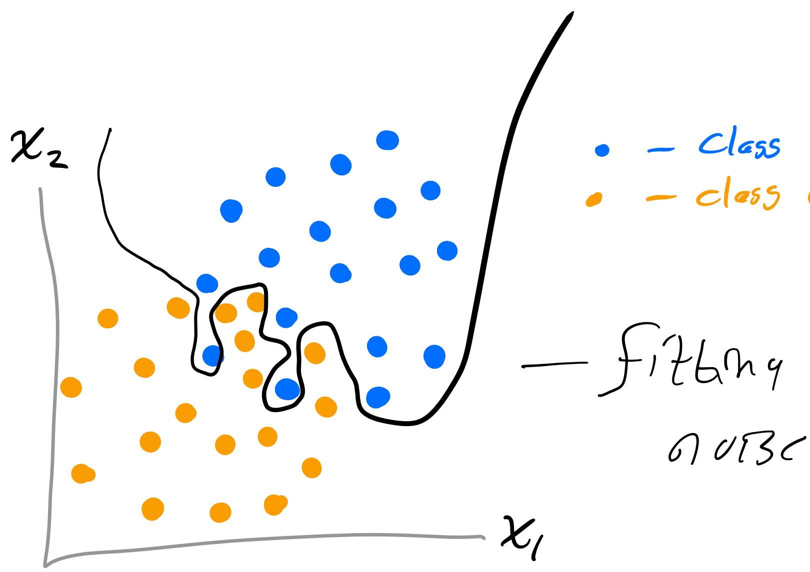
$\mathbb{R}^2$ $\mathbb{R}$

Training Data $\{x^{(i)}, y_i\}_{i=1\cdots n}$ $\quad y_i = \begin{cases} 1 & \text{if class } 1 \\ 0 & \text{if class } 0 \end{cases}$



Given this data, draw a decision boundary (curve where you would say class 1 is on one side and class 2 is on the other side)



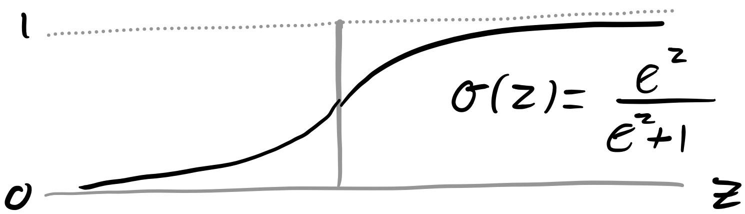


Model $$y = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2) = \hat{y}(x;\theta)$$

where

"Sigmoid"

"logistic function"



$$\sigma(z) = \frac{e^z}{e^z+1}$$

Solve $$\min_{\theta} \sum_{i=1}^{n} L(y_i, \hat{y}(x^{(i)};\theta)) \text{ for } \hat{\theta}$$

Predict: For new sample x, predict

$$\begin{cases} \text{class } 1 & \text{if } \hat{y} \geq \tfrac{1}{2} \\ \text{class } 0 & \text{if } \hat{y} < \tfrac{1}{2} \end{cases}$$

What loss function should you use?

One choice - log loss

$$L(\underbrace{y}_{\text{binary}}, \underbrace{\hat{y}}_{\text{continuous}}) = \begin{cases} -\log(\hat{y}) & \text{if } y=1 \\ -\log(1-\hat{y}) & \text{if } y=0 \end{cases}$$

$$= -y \log \hat{y} - (1-y)\log(1-\hat{y})$$

Decision Boundary for Logistic Regression

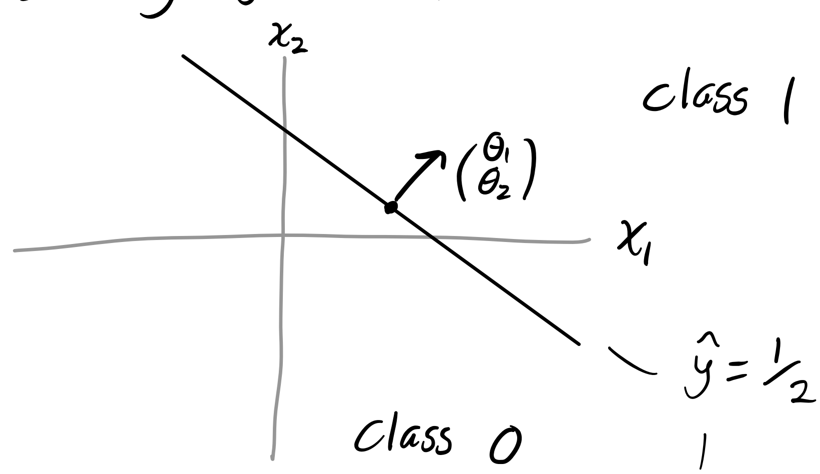
Training Data: $\{\underbrace{x^{(i)}}_{\mathbb{R}^2}, \underbrace{y_i}_{\mathbb{R}}\}_{i=1\ldots n}$ $\qquad y_i = \begin{cases} 1 & \text{if class } 1 \\ 0 & \text{if class } 0 \end{cases}$

Model: $$y = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2) = \hat{y}(x;\theta)$$

Predict: For new sample x, predict

$$\begin{cases} \text{class } 1 & \text{if } \hat{y} \geq \frac{1}{2} \\ \text{class } 0 & \text{if } \hat{y} < \frac{1}{2} \end{cases}$$

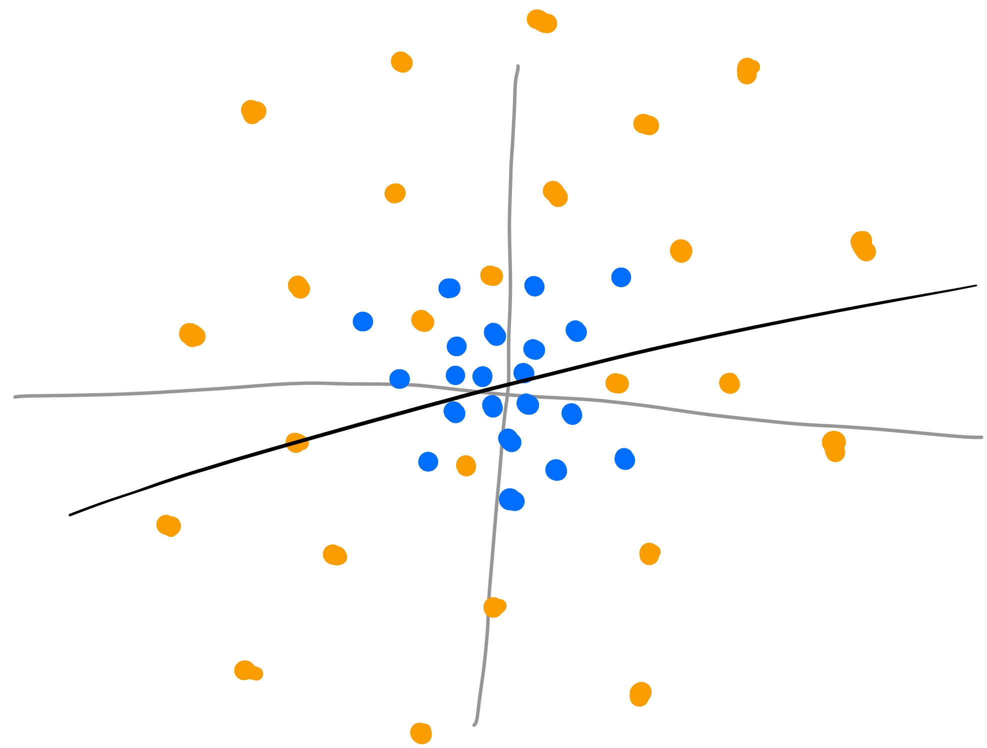
Decision boundary is linear



$$0 = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

Activity:

Could you use logistic regression to build a reasonable classifier for the following data?



$x^{(i)} \in \mathbb{R}^2$

a) $\hat{y} = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$?

Decision bdry is linear

Bad fit for this data.

b) $\hat{y} = \sigma\left(\theta_0 + \theta_1 \sqrt{x_1^2 + x_2^2}\right)$

Evaluating Classifiers

		Prediction +	Prediction −
Truth	+	True Positive (TP)	False Negative (FN)
	−	False Positive (FP)	True Negative (TN)

Accuracy: $\dfrac{TP + TN}{\text{total}}$

Activity: Someone invents a test for a rare disease that affects 0.1% of the population. The test has accuracy 99.9%. Are you convinced this is a good test?

Precision : $\frac{TP}{TP+FP}$ What fraction of predicted positives are real?

Recall : $\frac{TP}{TP+FN}$ What fraction of positives are correctly predicted?

Want high precision & high recall

Activity: You are building a binary classifier that detects whether a pedestrian is crossing the sidewalk within 30 feet of a self driving car. If the detection is positive, the car puts on the breaks. Would you rather have good precision and great recall or good recall and great precision?

There is a trade off between True Positives and False Positives, and between True Negatives and False Negatives

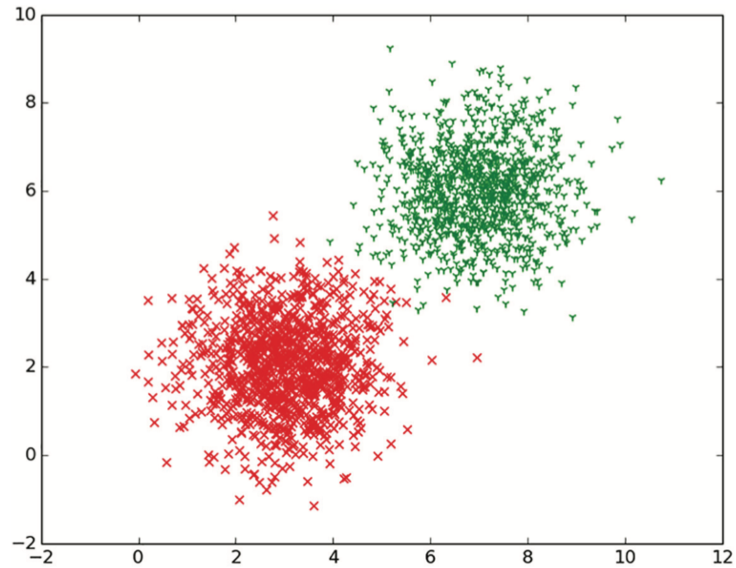
Training Data: $\{\underbrace{x^{(i)}}_{\mathbb{R}^2}, \underbrace{y_i}_{\mathbb{R}}\}_{i=1\ldots n}$ $\qquad y_i = \begin{cases} 1 & \text{if class } 1 \\ 0 & \text{if class } 0 \end{cases}$

Model: $y = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2) = \hat{y}(x;\theta)$

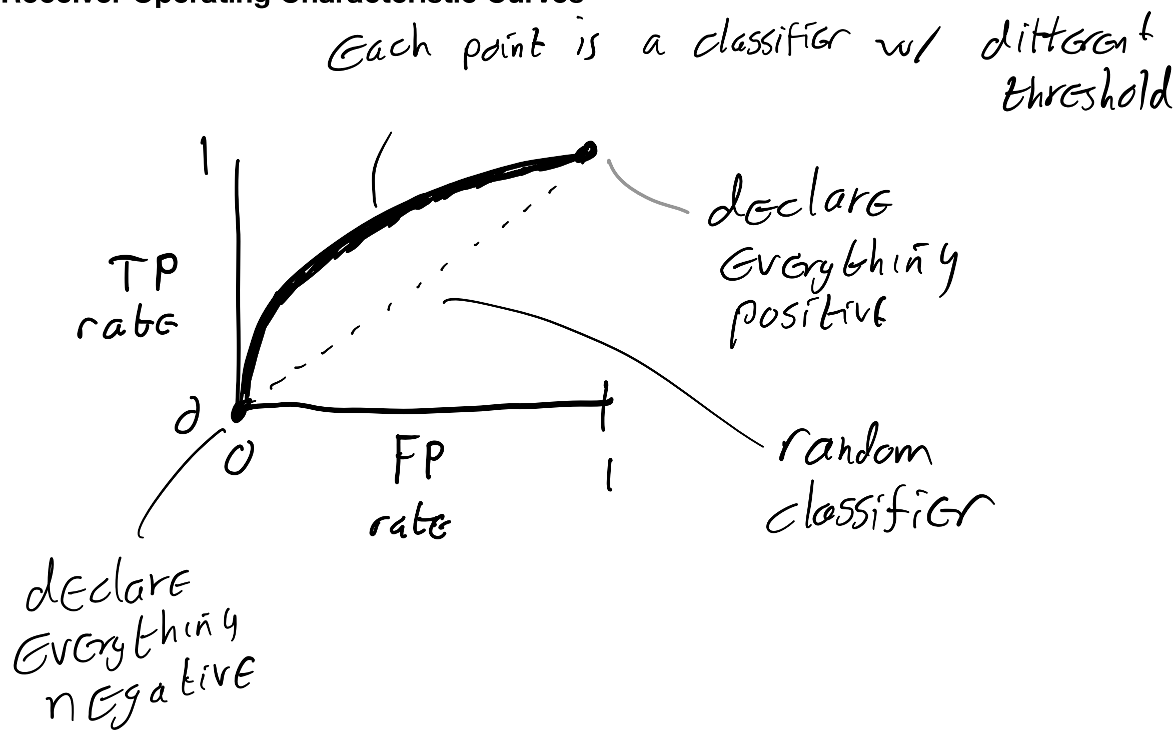
Predict: For new sample x, predict

$$\begin{cases} \text{class } 1 & \text{if } \hat{y} \geq \frac{1}{2} \\ \text{class } 0 & \text{if } \hat{y} < \frac{1}{2} \end{cases}$$

could choose any value



Receiver Operating Characteristic Curves



Comparing classifiers and Area-Under-Curve (AUC)

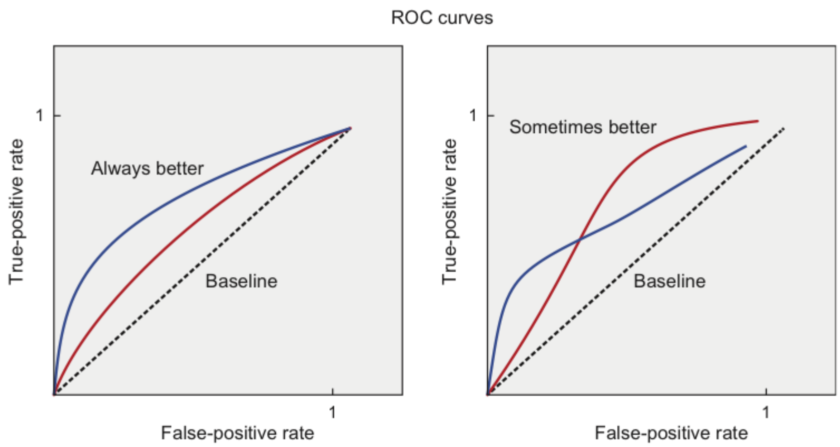
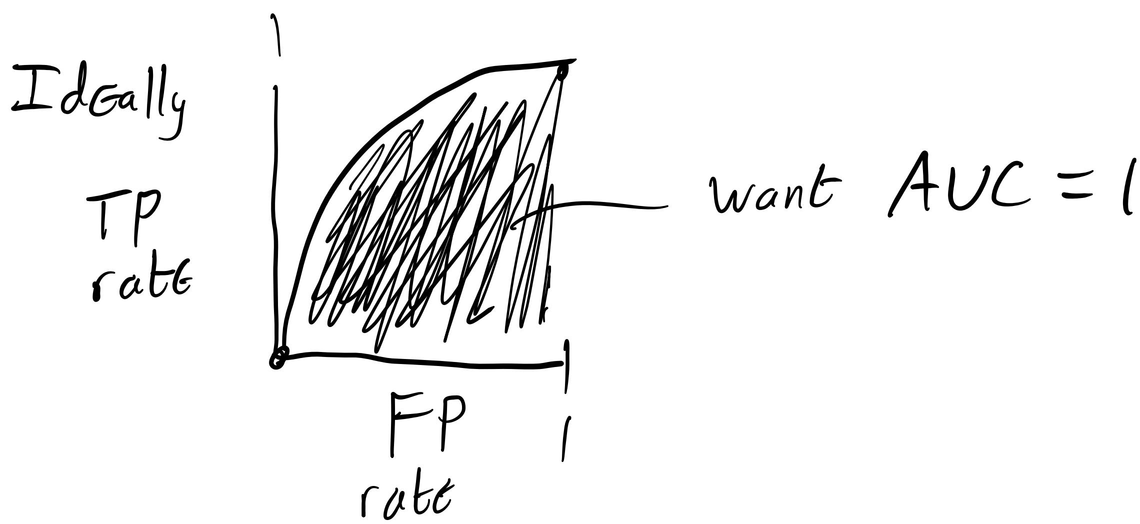


Figure 5.6 The principled way to compare algorithms is to examine their ROC curves. When the true-positive rate is greater than the false-positive rate in every situation, it's straightforward to declare that one algorithm is dominant in terms of its performance. If the true-positive rate is less than the false-positive rate, the plot dips below the baseline shown by the dotted line.



Also common to plot precision-recall curves