# Introduction to Machine Learning

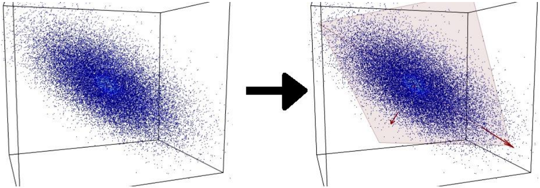
COMPSCI 371D — Machine Learning

# Outline

1. Unsupervised Learning
2. Drawings and Intuition in Higher Dimensions
3. Classification through Regression
4. Linear Separability

# Parenthesis: Supervised vs Unsupervised

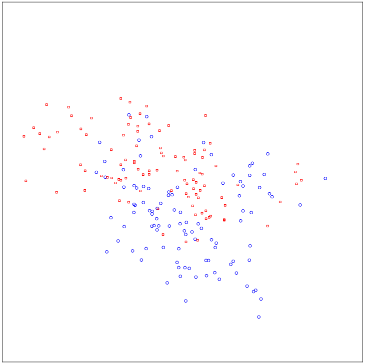
- Supervised: Train with $(\mathbf{x}, y)$
  - Classification: Hand-written digit recognition
  - Regression: Median age of YouTube viewers for each video
- Unsupervised: Train with $\mathbf{x}$
  - Clustering: Group customers by similar tastes to focus advertising
  - Dimensionality reduction: Which dimensions contain most of the variation?



[Image from cw.fel.cvut.cz]

- We will *not* cover unsupervised learning

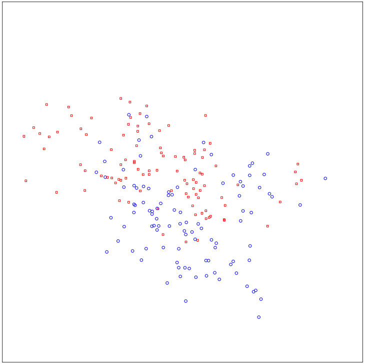
# Drawings Help Intuition



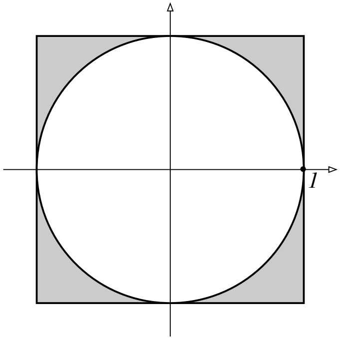
# Classifiers as Partitions of $X$

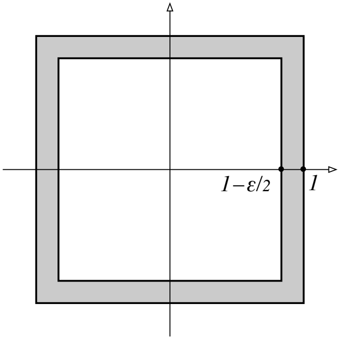
- $\hat{y} = h(x)$ for $\hat{y} \in Y$, a categorical set

  $X_{\hat{y}} \stackrel{\text{def}}{=} h^{-1}(\hat{y})$ *partitions* $X$ (not just $T$!)
- Classifier $=$ partition
- $S = h^{-1}$(red square), $C = h^{-1}$(blue circle)



# Intuition Often Fails in Many Dimensions



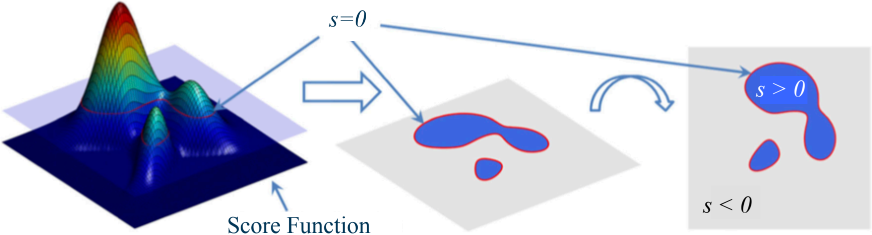




- Gray parts dominate when $d \to \infty$
- Distance from center to corners diverges when $d \to \infty$

# Classification and Geometry

- A classifier partitions $X \subset \mathbb{R}^d$ into sets, one per label in $Y$
- How do we represent sets $\subset \mathbb{R}^d$? How do we work with them?
- Simple partitions: logistic regression classifiers, linear support vector machines
- Complex partitions: nearest-neighbor classifier, kernel support vector machines, decision trees, neural networks
- Complex is not always better
- These methods have a strong geometric flavor
- Beware of our intuition!

# Score-Based Classifiers

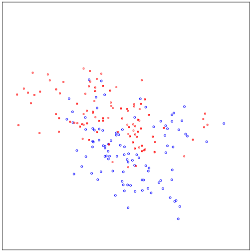



[Figure adapted from Wei *et al.*, *Structural and Multidisciplinary Optimization*, 58:831–849, 2018]

- $s = 0$ defines the *decision boundaries*
- $s > 0$ and $s < 0$ defines the (two) *decision regions*

# Score-Based Classifiers

- Threshold some *score function* $s(\mathbf{x})$:
- Example: `'s'`(red squares) and `'c'`(blue circles)



- Correspond to two sets $S \subseteq X$ and $C = X \setminus S$
  If we can estimate something like $s(\mathbf{x}) = \mathbb{P}[\mathbf{x} \in S]$

$$h(\mathbf{x}) = \begin{cases} \texttt{'s'} & \text{if } s(\mathbf{x}) > 1/2 \\ \texttt{'c'} & \text{otherwise} \end{cases}$$

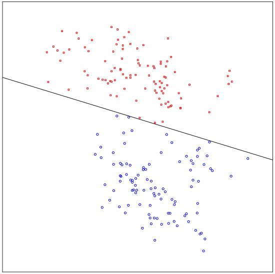
# Classification through Regression

- If you prefer 0 as a threshold, let
  $s(\mathbf{x}) = 2\mathbb{P}[\mathbf{x} \in S] - 1 \in [-1, 1]$

$$h(\mathbf{x}) = \begin{cases} \texttt{'s'} & \text{if } s(\mathbf{x}) > 0 \\ \texttt{'c'} & \text{otherwise} \end{cases}$$

- Scores are convenient even without probabilities, because they are easy to work with
- We implement a classifier $h$ by building a regressor $s$
- Example: Logistic-regression classifiers

# Linearly Separable Training Sets

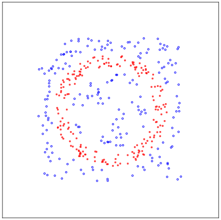


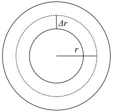
- Some line (hyperplane in $\mathbb{R}^d$) separates $C$, $S$
- Requires *much* smaller $\mathcal{H}$
- Simplest score: $s(\mathbf{x}) = b + \mathbf{w}^T\mathbf{x}$. The line is $s(\mathbf{x}) = 0$

$$h(\mathbf{x}) = \begin{cases} \texttt{'s'} & \text{if } s(\mathbf{x}) > 0 \\ \texttt{'c'} & \text{otherwise} \end{cases}$$

# Data Representation?

- Linear separability is a property of the data *in a given representation*






- Xform 1: $z = x_1^2 + x_2^2$ implies $\mathbf{x} \in S \Leftrightarrow a \leq z \leq b$
- Xform 2: $u = |\sqrt{x_1^2 + x_2^2} - r| = |\sqrt{z} - r|$
  yields linear separability:
  $\mathbf{x} \in S \Leftrightarrow u \leq \Delta r$