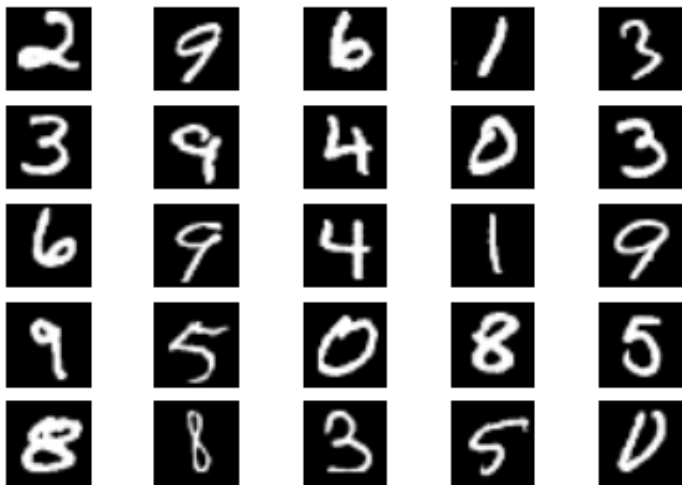


# CSC 411: Lecture 03: Linear Classification

Richard Zemel, Raquel Urtasun and Sanja Fidler

University of Toronto

# Examples of Classification



What digit is this?

# Examples of Classification



Is this a dog?

# Examples of Classification



what about this one?

# Examples of Classification



Am I going to pass the exam?

# Examples of Classification



Do I have diabetes?

- What do all these problems have in common?
- Categorical **outputs**, called **labels**  
(eg, yes/no, dog/cat/person/other)
- Assigning each input vector to one of a finite number of labels is called **classification**
- **Binary classification**: two possible labels (eg, yes/no, 0/1, cat/dog)
- **Multi-class classification**: multiple possible labels
- We will first look at binary problems, and discuss multi-class problems later in class

- Linear Classification (binary)
- Key Concepts:
  - ▶ Classification as regression
  - ▶ **Decision boundary**
  - ▶ **Loss** functions
  - ▶ **Metrics** to evaluate classification



# Classification vs Regression

- We are interested in mapping the input  $\mathbf{x} \in \mathcal{X}$  to a *label*  $t \in \mathcal{Y}$
- In regression typically  $\mathcal{Y} = \mathbb{R}$
- Now  $\mathcal{Y}$  is categorical

# Classification as Regression

- Can we do this task using what we have learned in previous lectures?
- Simple hack: Ignore that the output is categorical!
- Suppose we have a binary problem,  $t \in \{-1, 1\}$
- Assuming the standard model used for (linear) regression

$$y(\mathbf{x}) = f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x}$$

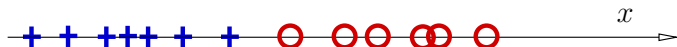
- How can we obtain  $\mathbf{w}$ ?
- Use least squares,  $\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}$ . How is  $\mathbf{X}$  computed? and  $\mathbf{t}$ ?
- Which loss are we minimizing? Does it make sense?

$$\ell_{square}(\mathbf{w}, t) = \frac{1}{N} \sum_{n=1}^N (t^{(n)} - \mathbf{w}^T \mathbf{x}^{(n)})^2$$

- How do I compute a label for a new example? Let's see an example

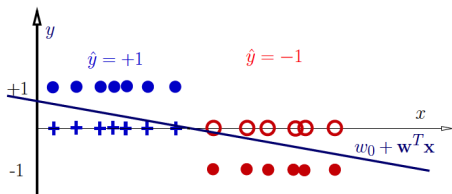
# Classification as Regression

- One dimensional example (input  $x$  is 1-dim)



- The colors indicate labels (a blue plus denotes that  $t^{(i)}$  is from the first class, red circle that  $t^{(i)}$  is from the second class)

# Decision Rules



- Our classifier has the form

$$f(\mathbf{x}, \mathbf{w}) = w_0 + \mathbf{w}^T \mathbf{x}$$

- A reasonable **decision rule** is

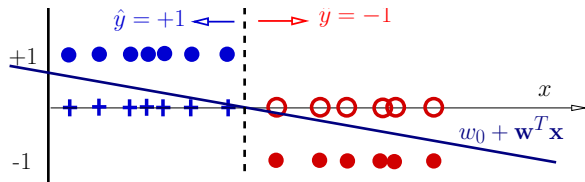
$$y = \begin{cases} 1 & \text{if } f(\mathbf{x}, \mathbf{w}) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

- How can I mathematically write this rule?

$$y(\mathbf{x}) = \text{sign}(w_0 + \mathbf{w}^T \mathbf{x})$$

- What does this function look like?

# Decision Rules



- How can I mathematically write this rule?

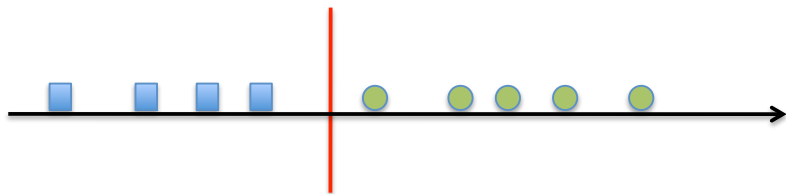
$$y(\mathbf{x}) = \text{sign}(w_0 + \mathbf{w}^T \mathbf{x})$$

- This specifies a **linear classifier**: it has a **linear boundary (hyperplane)**

$$w_0 + \mathbf{w}^T \mathbf{x} = 0$$

which separates the space into two "half-spaces"

# Example in 1D



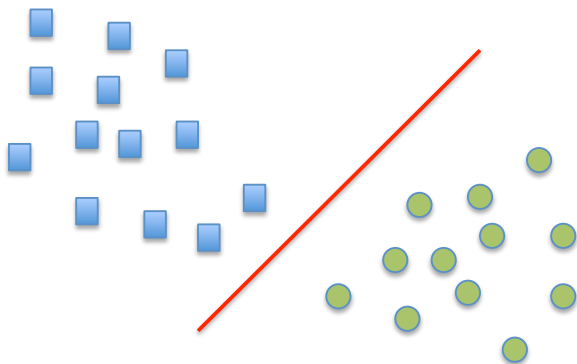
- The **linear classifier** has a **linear boundary (hyperplane)**

$$w_0 + \mathbf{w}^T \mathbf{x} = 0$$

which separates the space into two "half-spaces"

- In 1D this is simply a threshold

## Example in 2D



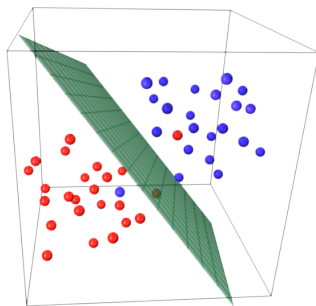
- The **linear classifier** has a **linear boundary (hyperplane)**

$$w_0 + \mathbf{w}^T \mathbf{x} = 0$$

which separates the space into two "half-spaces"

- In 2D this is a line

## Example in 3D



- The **linear classifier** has a **linear boundary (hyperplane)**

$$w_0 + \mathbf{w}^T \mathbf{x} = 0$$

which separates the space into two "half-spaces"

- In 3D this is a plane
- What about higher-dimensional spaces?



# Geometry

$\mathbf{w}^T \mathbf{x} = 0$  a line passing through the origin and orthogonal to  $\mathbf{w}$   
 $\mathbf{w}^T \mathbf{x} + w_0 = 0$  shifts it by  $w_0$

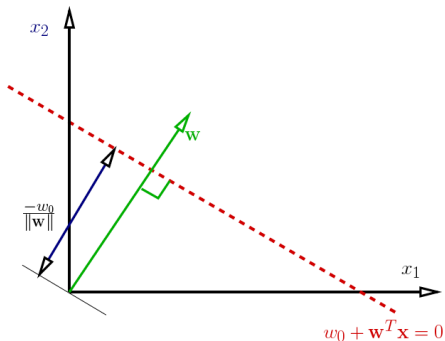
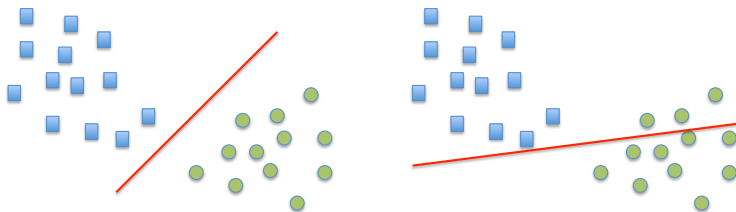


Figure from G. Shakhnarovich

# Learning Linear Classifiers

- Learning consists in estimating a “good” decision boundary
- We need to find  $\mathbf{w}$  (direction) and  $w_0$  (location) of the boundary
- What does “good” mean?
- Is this boundary good?



- We need a criteria that tell us how to select the parameters
- Do you know any?

- Classifying using a linear decision boundary reduces the data dimension to 1

$$y(\mathbf{x}) = \text{sign}(w_0 + \mathbf{w}^T \mathbf{x})$$

- What is the cost of being wrong?
- **Loss function:**  $L(y, t)$  is the loss incurred for predicting  $y$  when correct answer is  $t$
- For medical diagnosis: For a diabetes screening test is it better to have false positives or false negatives?
- For movie ratings: The "truth" is that Alice thinks E.T. is worthy of a 4. How bad is it to predict a 5? How about a 2?

- A possible loss to minimize is the **zero/one loss**

$$L(y(\mathbf{x}), t) = \begin{cases} 0 & \text{if } y(\mathbf{x}) = t \\ 1 & \text{if } y(\mathbf{x}) \neq t \end{cases}$$

- Is this minimization easy to do? Why?

# Other Loss functions

- Zero/one loss for a classifier

$$L_{0-1}(y(\mathbf{x}), t) = \begin{cases} 0 & \text{if } y(\mathbf{x}) = t \\ 1 & \text{if } y(\mathbf{x}) \neq t \end{cases}$$

- Asymmetric Binary Loss

$$L_{ABL}(y(\mathbf{x}), t) = \begin{cases} \alpha & \text{if } y(\mathbf{x}) = 1 \wedge t = 0 \\ \beta & \text{if } y(\mathbf{x}) = 0 \wedge t = 1 \\ 0 & \text{if } y(\mathbf{x}) = t \end{cases}$$

- Squared (quadratic) loss

$$L_{squared}(y(\mathbf{x}), t) = (t - y(\mathbf{x}))^2$$

- Absolute Error

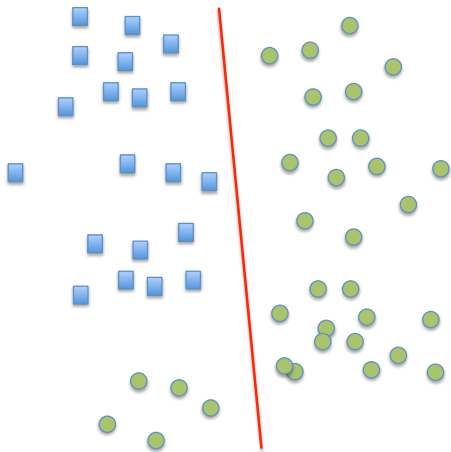
$$L_{absolute}(y(\mathbf{x}), t) = |t - y(\mathbf{x})|$$

# More Complex Loss Functions

- What if the movie predictions are used for rankings? Now the predicted ratings don't matter, just the order that they imply.
- In what order does Alice prefer E.T., Amelie and Titanic?
- Possibilities:
  - ▶ 0-1 loss on the winner
  - ▶ Permutation distance
  - ▶ Accuracy of top K movies.

# Can we always separate the classes?

- If we can separate the classes, the problem is **linearly separable**



# Can we always separate the classes?

Causes of non perfect separation:

- Model is too simple
- Noise in the inputs (i.e., data attributes)
- Simple features that do not account for all variations
- Errors in data targets (mis-labelings)

Should we make the model complex enough to have perfect separation in the training data?



# Metrics

How to evaluate how good my classifier is? How is it doing on dog vs no-dog?



— TP (True Positive)

— FP (False Positive)

— FN (False Negative)

How to evaluate how good my classifier is?

- **Recall:** is the fraction of relevant instances that are retrieved

$$R = \frac{TP}{TP + FN} = \frac{TP}{\text{all groundtruth instances}}$$

- **Precision:** is the fraction of retrieved instances that are relevant

$$P = \frac{TP}{TP + FP} = \frac{TP}{\text{all predicted}}$$

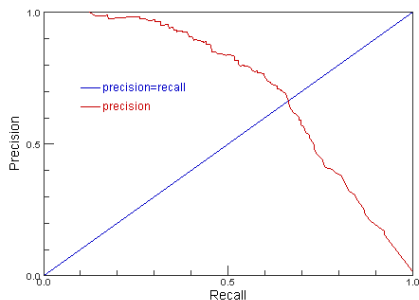
- **F1 score:** harmonic mean of precision and recall

$$F1 = 2 \frac{P \cdot R}{P + R}$$

# More on Metrics

How to evaluate how good my classifier is?

- **Precision:** is the fraction of retrieved instances that are relevant
- **Recall:** is the fraction of relevant instances that are retrieved
- **Precision Recall Curve**



- **Average Precision (AP):** mean under the curve

# Metrics vs Loss

- Metrics on a dataset is what we care about (performance)
- We typically cannot directly optimize for the metrics
- Our loss function should reflect the problem we are solving. We then hope it will yield models that will do well on our dataset