Models of Classification

Machine Learning Techniques

Dr. Ashish Tendulkar
IIT Madras

What topics will be discussed?

1. Discriminant functions: Learn direct mapping between feature vector $\mathbf{x}$ and label y.
2. Generative and discriminative models:
    - Generative classifiers model class conditional densities $p(\mathbf{x}|y)$ for features and prior probabilities of classes $p(y)$ and then through Bayes theorem, calculate $p(y|\mathbf{x})$.
    - Discriminative classifiers learn conditional probability distribution $p(y|\mathbf{x})$ through parameteric models.
3. Instance based models - Compare the test examples with the training examples and assigns class labels based on certain measure of similarity.

Part I: Classification setup

Classification set up

- Predict class label y of an example based on the feature vector $\mathbf{x}$.
- Class label y is a discrete quantity unlike a real number in regression set up.

Nature of class label

- Label is a discrete quantity - precisely an element in some finite set of class labels.
- Depending on the nature of the problem, we have one or more labels assigned to each example.

Types of classification

1. **Single label classification** - where each example has exactly one label.
   - e.g. is *the applicant eligible for loan?*
   - Label set: {yes, no}.
   - Label either *yes* or *no*.
2. **Multi-label classification** - where each example has more than one label.
   - e.g. identifying different types of fruits in a picture.

Label representation: Single example

1. **Single label classification:** Label is a scalar quantity and is represented by y.
2. **Multi-label classification:** More than one label hence vector representation $\mathbf{y}$.

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}$$

Label set: $y = \{y_1, y_2, \ldots, y_k\}$ has k elements/labels.

Depending on the presence of the label, the corresponding label is set to 1.

Example: Single label classification (Binary)

- *Is the application eligible for loan?*
- Label set: $\{yes, no\}$, usually converted to $\{1, 0\}$
  - Label: either yes (1) or $no(0)$.
- Training example:
  - Feature vector: $\mathbf{x}$ - features for loan application like *age of applicant, income, number of dependents* etc.
  - Label: y

Example: Single label classification (Multiclass)

- *Types of iris flower*
- Label set: $C = \{versicolor, setosa, virginica\}$
- Label: exactly one label from set C.

Example: Single label classification (Multiclass)

- *Types of iris flower*

versicolor

setosa

virginica

Image Source: Wikipedia.org

Label encoding in multiclass setup

Use one-hot encoding scheme for label encoding.

- Make use of a vector $\mathbf{y}$ with components equal to the number of labels in the label set.
- In iris example, this would become:

$$\mathbf{y} = \begin{bmatrix} y_{versicolor} \\ y_{setosa} \\ y_{virginica} \end{bmatrix}$$

Example: Label encoding (single label)

Let's assume that the flower has label *versicolor*, we will encode it as follows:

$$\mathbf{y} = \begin{bmatrix} y_{versicolor} = 1 \\ y_{setosa} = 0 \\ y_{virginica} = 0 \end{bmatrix}$$

Note that the component of $\mathbf{y}$ corresponding to the label *versicolor* is 1, every other component is 0.

$$\mathbf{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Example: Multi-label Classification

- *Label all fruits from an image*.
- Label set: List of fruits e.g.

  $\{apple, guava, mango, orange, banana, strawberry, \}$
- Label: One or more fruits as they are present in the image.

$$\mathbf{y} = \begin{bmatrix} y_{apple} \\ y_{guava} \\ y_{mango} \\ y_{orange} \\ y_{banana} \\ y_{strawberry} \end{bmatrix}$$

Example: Multi-label Classification

Sample image

Image source: Wikipedia.org

Example: Multi-label Classification

Different fruits in the images are:

Apple

Orange

Banana

Example: Multi-label Classification

- Let's assume that the labels are **apple**, **orange** and **banana**.

$$\mathbf{y} = \begin{bmatrix} y_{apple} = 1 \\ y_{guava} = 0 \\ y_{mango} = 0 \\ y_{orange} = 1 \\ y_{banana} = 1 \\ y_{strawberry} = 0 \end{bmatrix} \quad \text{becomes} \quad \mathbf{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Training Data: Binary Classification

- Let's denote D as a set of n pairs of a features vector $\mathbf{x}_{m \times 1}$ and a label y, to represent examples.

$$D = \{(\mathbf{X}, \mathbf{y})\} = \left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{n}$$

- $\mathbf{X}$ is a feature matrix corresponding to all the training examples and has shape $n \times m$. In this matrix, each feature vector is transposed and represented as a row in this matrix.

Training Data: Binary Classification

$$D = \{(\mathbf{X}, \mathbf{y})\} = \left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{n}$$

Concretely, the feature vector for i-th training example $\mathbf{x}^{(i)}$ can be obtained by $\mathbf{X}[i]$:

$$\mathbf{X}_{n\times m} = \begin{bmatrix} - \left(x^{(1)}\right)^T - \\ - \left(x^{(2)}\right)^T - \\ \vdots \\ - \left(x^{(n)}\right)^T - \end{bmatrix}$$

Training Data: Binary Classification

$$D = \{(\mathbf{X}, \mathbf{y})\} = \left\{(\mathbf{x}^{(i)}, y^{(i)})\right\}_{i=1}^{n}$$

$\mathbf{y}$ is a label vector of shape $n \times 1$. The i-th entry in this vector gives label for i-th example, which is denoted by $y^{(i)}$.

$$\mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

Training Data: Multi-class classification

A set of n pairs of a feature vector $\mathbf{x}$ and a label vector $\mathbf{y}$ representing examples.

We denote it by D:

$$D = \{(\mathbf{X}, \mathbf{Y})\} = \left\{\left(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\right)\right\}_{i=1}^{n}$$

where

$\mathbf{X}$ is an $n \times m$ feature matrix:

$$\mathbf{X}_{n\times m} = \begin{bmatrix} - \left(\mathbf{x}^{(1)}\right)^T - \\ - \left(\mathbf{x}^{(2)}\right)^T - \\ \vdots \\ - \left(\mathbf{x}^{(n)}\right)^T - \end{bmatrix}$$

Training Data: Multi-class classification

$$D = \{(\mathbf{X}, \mathbf{Y})\} = \left\{\left(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\right)\right\}_{i=1}^{n}$$

$\mathbf{Y}$ is a label matrix of shape $n \times k$, where k is the total number of classes in label set.

$$\mathbf{Y} = \begin{bmatrix} - \left(\mathbf{y}^{(1)}\right)^T - \\ - \left(\mathbf{y}^{(2)}\right)^T - \\ \vdots \\ - \left(\mathbf{y}^{(n)}\right)^T - \end{bmatrix}$$

Multi-class and multi-label classification label vector

$\mathbf{Y}$ is a label matrix of shape $n \times k$, where k is the total number of classes in label set.

$$\mathbf{Y} = \begin{bmatrix} - \left(\mathbf{y}^{(1)}\right)^T - \\ - \left(\mathbf{y}^{(2)}\right)^T - \\ \vdots \\ - \left(\mathbf{y}^{(n)}\right)^T - \end{bmatrix}$$

Multi-class and multi-label classification label vector

- Multi-class classification: For $\left(\mathbf{y}^{(i)}\right)^T$, **exactly one entry** corresponding to the class label is 1.
- Multi-label classification: For $\left(\mathbf{y}^{(i)}\right)^T$, **more than one entries** corresponding to the class labels can be 1.

Part II: Discriminant Functions

Overview

Example: Two classes

Simplest discriminant function is very similar to the linear regression:

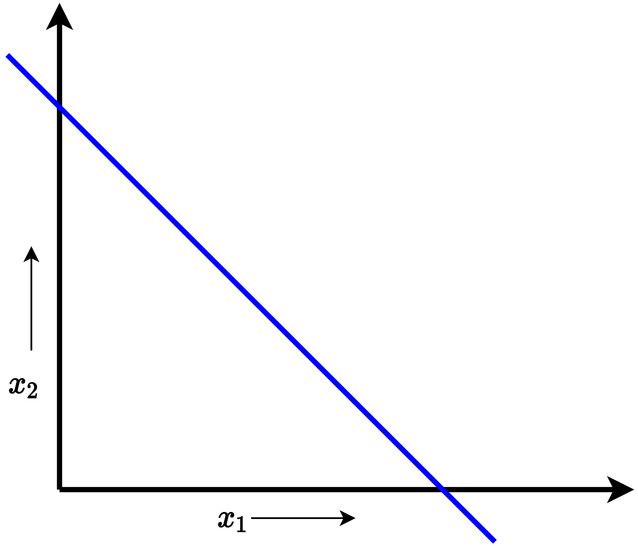
$$
\begin{aligned}
y &= w_0 + w_1 x_1 + \ldots + w_m x_m \\
&= w_0 + \mathbf{w}^T \mathbf{x}
\end{aligned}
$$

where,

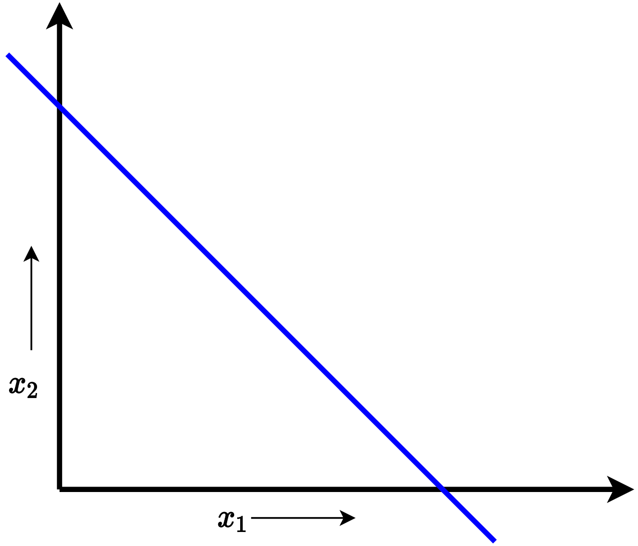
- w_0: Bias [*Keeping this separately for a reason*]
- $\mathbf{w}$: Weight vector
- $\mathbf{x}$: Feature vector
- y: label

Geometric Interpretation

The simplest discriminant function $y = w_o + \mathbf{w}^T\mathbf{x}$ represents a hyperplane in $m - 1$ dimensional space where m is the number of features.

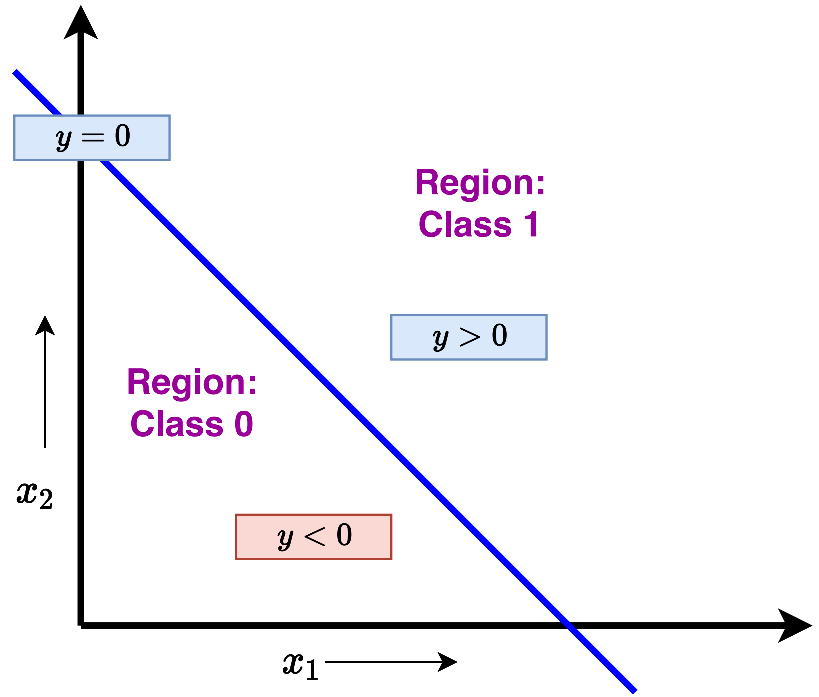


Geometric Interpretation

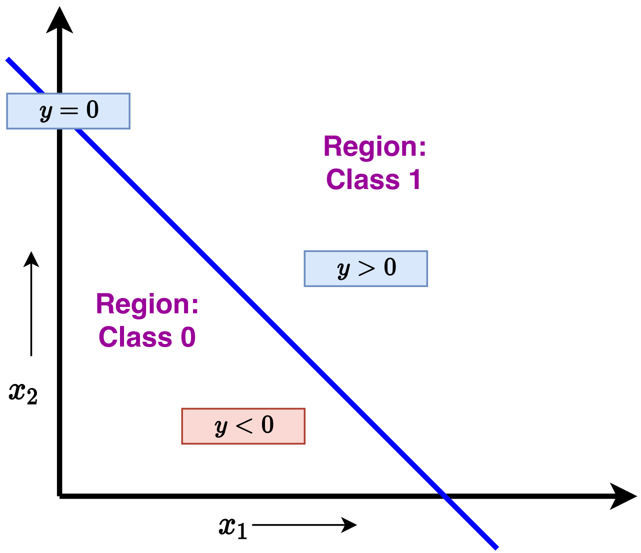


The discriminant function is a hyperplane in (m-1)-D space i.e. $2 - 1 = 1$-D space, which is a line. Note that here $m = 2$ features.

Classification with discriminant functions



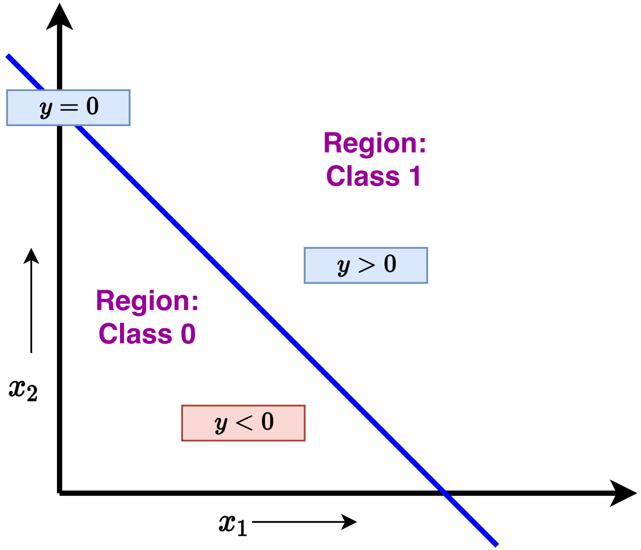
Classification with discriminant functions



Classification is performed as follows:

$$y = \begin{cases} 1, \text{ if } w_0 + \mathbf{w}^T \mathbf{x} > 0 \\ 0, \text{ otherwise} \end{cases}$$

Classification with discriminant functions



The decision boundary is defined by

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

What does $\mathbf{w}$ represent?

Consider two points $x^{(A)}$ and $x^{(B)}$ on the decision surface, we will have

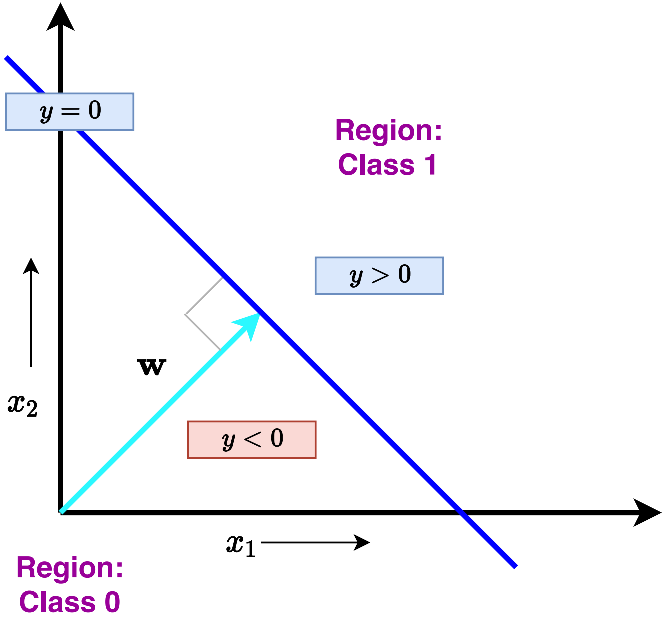
$$y^{(A)} = w_0 + \mathbf{w}^T \mathbf{x}^{(A)} = 0$$
$$y^{(B)} = w_0 + \mathbf{w}^T \mathbf{x}^{(B)} = 0$$

Since $y^{(A)} = y^{(B)} = 0$, $y^{(A)} - y^{(B)}$ results into the following equation:

$$\mathbf{w}^T (\mathbf{x}^{(A)} - \mathbf{x}^{(B)}) = 0$$

What does $\mathbf{w}$ represent?

The vector $\mathbf{w}$ is orthogonal to every vector lying within the decision surface, hence it determines the **orientation of the decision surface**.



What does w_0 represent?

For points on decision surface, we have

$$
\begin{aligned}
w_0 + \mathbf{w}^T \mathbf{x} &= 0 \\
\mathbf{w}^T \mathbf{x} &= -w_0
\end{aligned}
$$

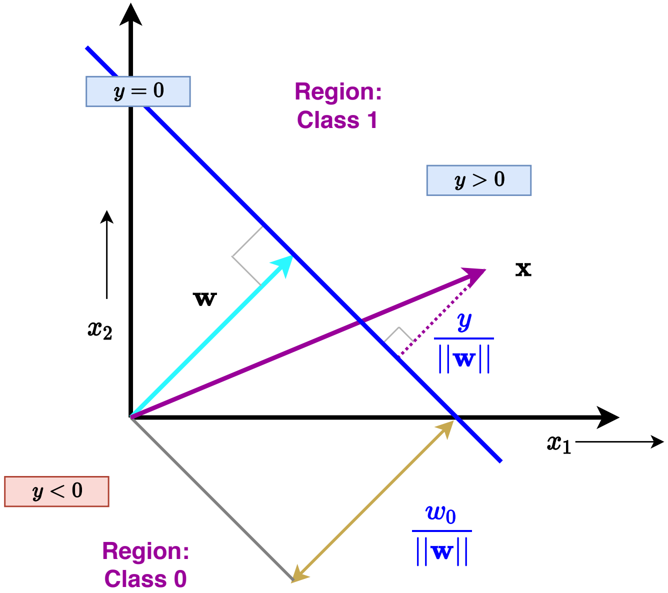
What does w_0 represent?

Normalizing both sides with the length of the vector $||\mathbf{w}||$, we get normal distance from the origin to the decision surface:

$$\frac{\mathbf{w}^T\mathbf{x}}{||\mathbf{w}||} = -\frac{w_0}{||\mathbf{w}||}$$

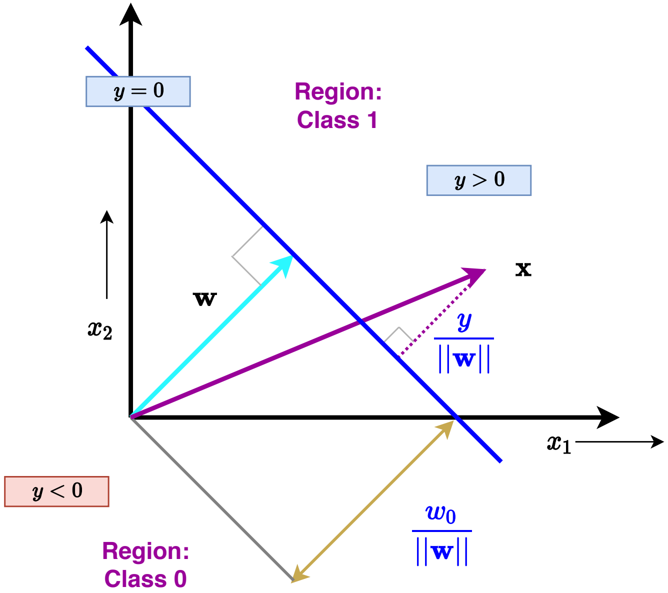
$\boldsymbol{w_0}$ determines the location of the decision surface

What does y represent?



y gives signed measure of perpendicular distance of the point $\mathbf{x}$ from the decision surface.

What does y represent?



- w_0 determines the **location** of the decision surface.
- $\mathbf{w}$ is orthogonal to every vector lying within the decision surface, hence it determines the **orientation** of the decision surface.

Alternate interpretation

By using a dummy feature x_0 and setting it to 1, we get the following equation:

$$\begin{aligned} y &= w_0x_0 + w_1x_1 + \ldots + w_mx_m \\ &= \mathbf{w}^T\mathbf{x} \end{aligned}$$

This represents a decision surface that is m-D hyperplane passing through the origin of $(m+1)$-D space.

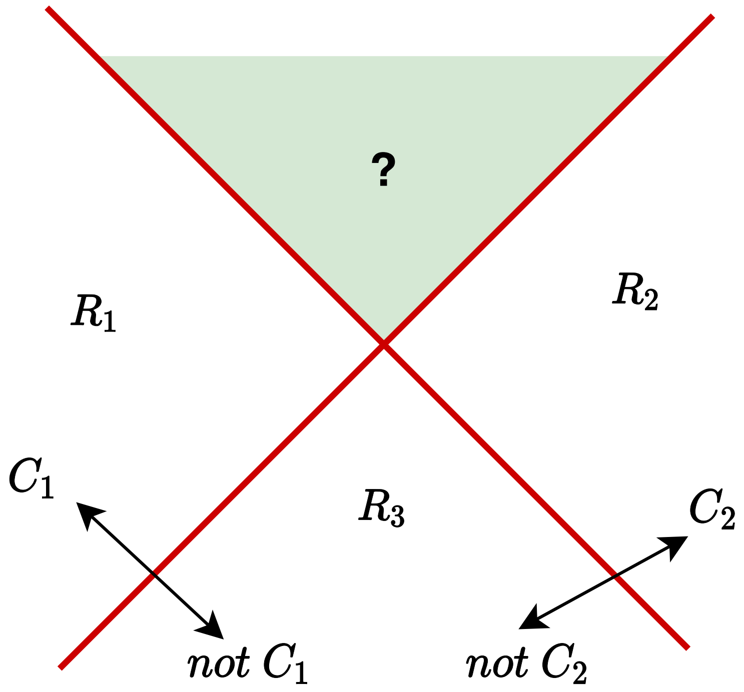
Multiple classes

Assuming the number of classes to be $k > 2$, we can build discriminant functions in two ways:

- **One-vs-rest**: Build $k - 1$ discriminant functions. Each discriminant function solves two class classification problem: class C_k vs $not\ C_k$.
- **One-vs-one**: One discriminant function per pair of classes. Total functions = $\binom{k}{2} = \frac{k\,(k-1)}{2}$

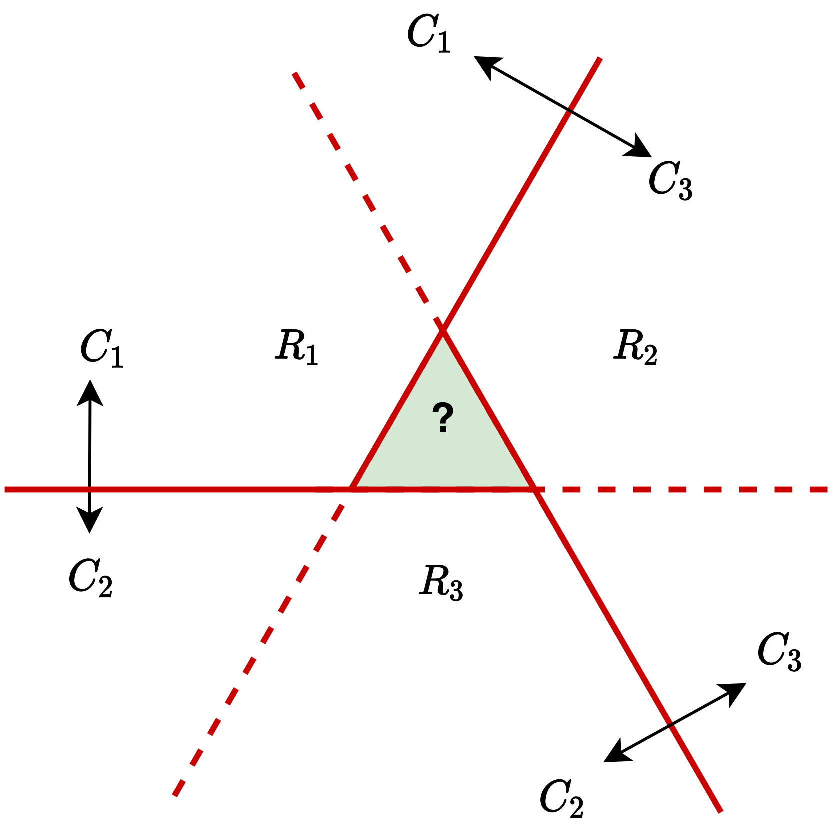
Issues with *one-vs-rest*

- Two discriminant functions for each class C_1 and C_2.
- Each discriminant function separates C_k and not C_k.
- Region of ambiguity is in green.



Issues with *one-vs-one*

- $k(k-1)/2$ discriminant functions for each class pair C_i and C_j.
- Each discriminant function separates C_i and C_j.
- Each point is classified by majority vote.
- Region of ambiguity is in green.



How do we fix it?

A single k-class discriminant comprising k linear functions as follows:

$$
\begin{aligned}
y_k &= w_{k0} + w_{k1}x_1 + \ldots + w_{km}x_m \\
&= w_{k0} + \mathbf{w_k}^T\mathbf{x}
\end{aligned}
$$

How do we fix it?

Concretely:

$$
\begin{aligned}
y_1 &= \quad w_{10} + \mathbf{w_1}^T \mathbf{x} \\
y_2 &= \quad w_{20} + \mathbf{w_2}^T \mathbf{x} \\
&\vdots \\
y_k &= \quad w_{k0} + \mathbf{w_k}^T \mathbf{x}
\end{aligned}
$$

Classification in k-discriminant functions

Assign label y_k to example $\mathbf{x}$ if $y_k > y_j, \forall j \neq k$

The decision boundary between classes y_k and y_j corresponds to $m-1$ dimensional hyperplane:

$$(w_{k0} - w_{j0}) + (\mathbf{w}_k - \mathbf{w}_j)^T \mathbf{x} = 0$$

This has the same form as the decision boundary for the two class cases:

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

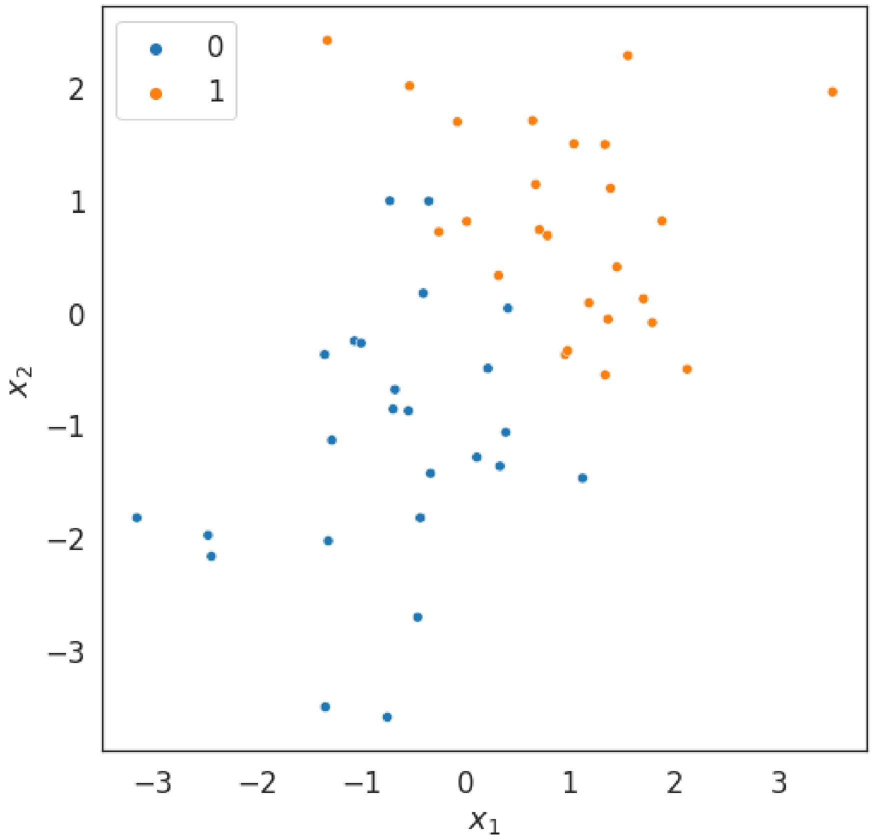
Now that we have a model of linear discriminant functions, we will study two approaches for learning the parameters of the model:

- Least squares
- Perceptron

Least squares classification

Train-test split (TODO)

Sample Training Data



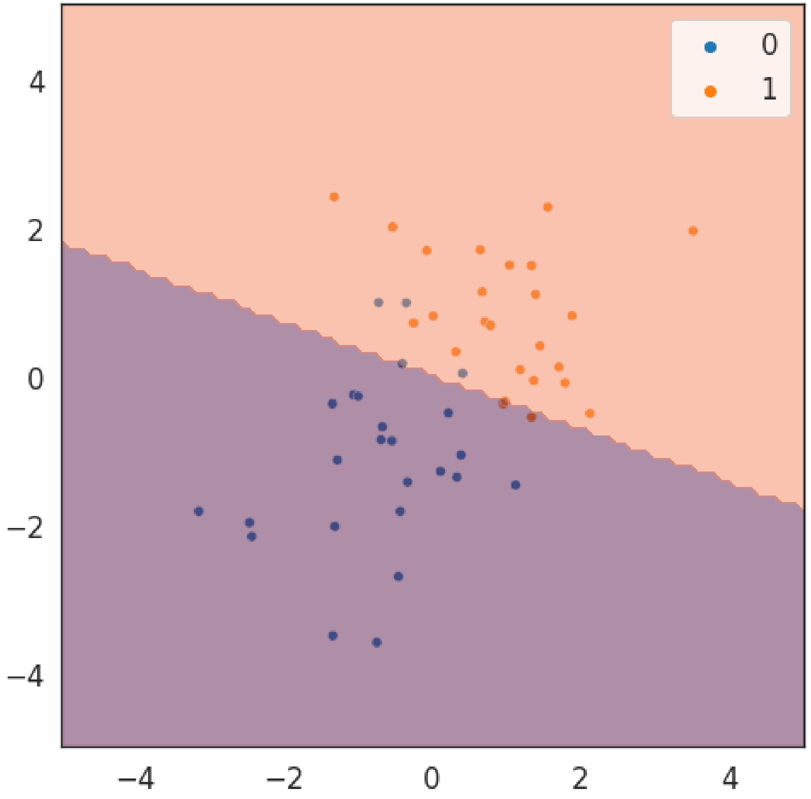
Sample Training Data

Let's implement the model inference function:

```python
def predict(x, w):
  z = x @ w
  return np.array([1 if z_val >= 0 else 0 for z_val in z])
```

Decision Boundary Visualization

A random decision boundary



Loss function: Least Square Error

The total loss is the sum of square of errors between actual and predicted labels at each training point.

The error at i-th training point is calculated as follows:

$$\begin{aligned} e^{(i)} &= (\text{actual label} - \text{predicted label})^2 \\ &= \left(y^{(i)} - h_{\mathbf{w}}(\mathbf{x}^{(i)})\right)^2 \\ &= \left(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)}\right)^2 \end{aligned}$$

Loss function: Least Square Error

The total loss $J(\mathbf{w})$ is sum of errors for each training point:

$$J(\mathbf{w}) = \sum_{i=1}^{n} e^{(i)} = \mathbf{e}^T \mathbf{e}$$

Note that the loss is dependent on the value of $\mathbf{w}$ - as these value changes, we get a new model, which will result in different prediction and hence affects the error at each training point.

Optimization: Normal equation

Calculate derivative of loss function $J(\mathbf{w})$ w.r.t. weight vector $\mathbf{w}$.

$$\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}} = 2(\mathbf{X}^T\mathbf{X}\mathbf{W} - \mathbf{X}^T\mathbf{Y})$$

Set $\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}}$ to 0 and solve for $\mathbf{W}$:

$$0 = 2(\mathbf{X}^T\mathbf{X}\mathbf{W} - \mathbf{X}^T\mathbf{Y})$$
$$\mathbf{W} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y}$$

Optimization: Normal equation

$$\mathbf{W} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y}$$

Whenever $\mathbf{X}^T\mathbf{X}$ is not full rank, we calculate pseudo-inverse:
$\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$

Evaluation metrics

- Confusion matrix
- Precision/Recall/F1 score

Note to Swarnim: Please write one line code for confusion matrix and precision/recall/f1 and report these metrics in a slide: on one side show confusion matrix and on the other side all metric values.