CE9010: Introduction to Data Science

Lecture 4: Supervised Classification

Semester 2 2017/18

Xavier Bresson

School of Computer Science and Engineering
Data Science and AI Research Centre
Nanyang Technological University (NTU), Singapore

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Outline

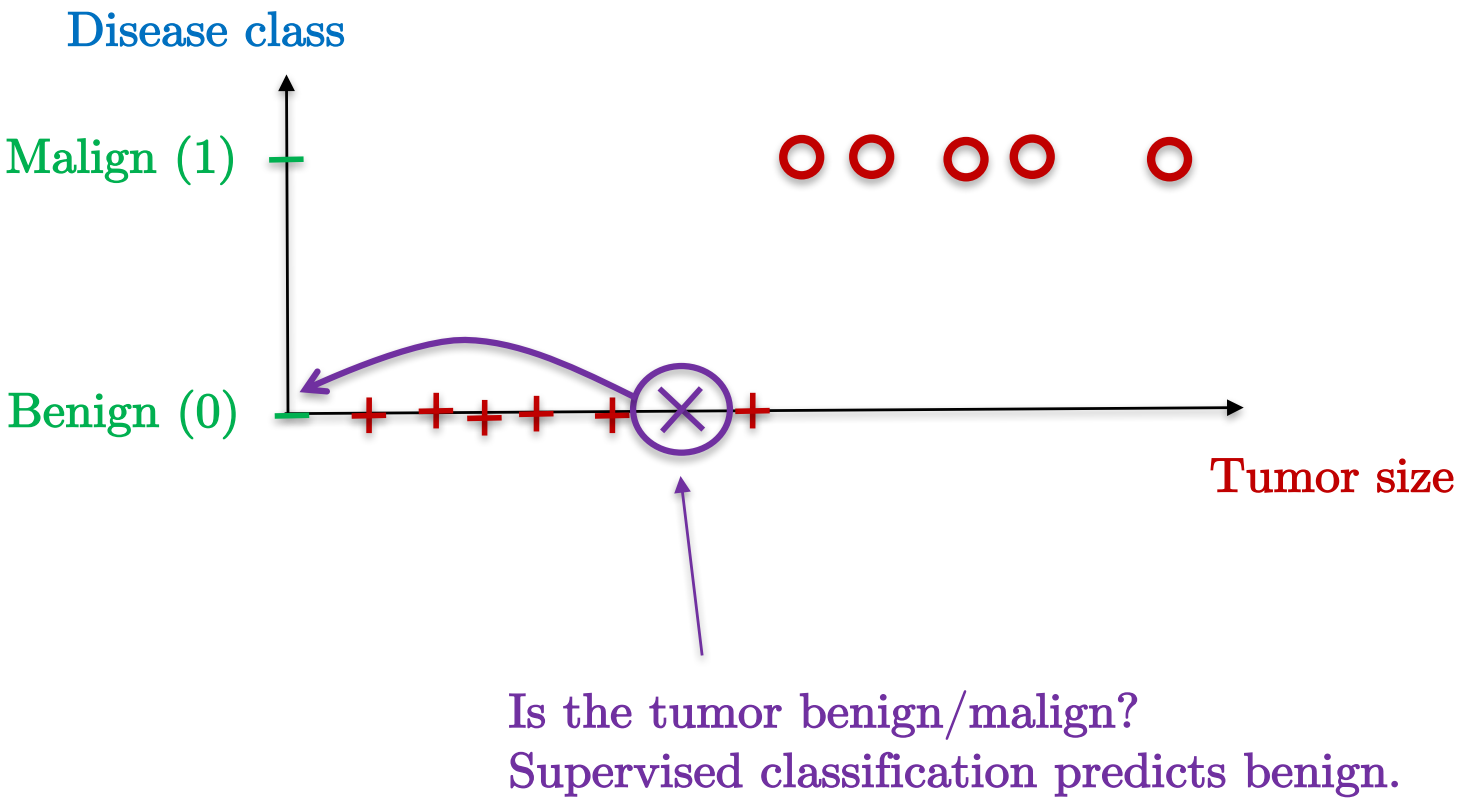
- Classification problem
- Predictive function
- Decision boundary
- Loss function
- Gradient descent
- Multi-class classification
- Conclusion

Outline

- Classification problem
- Predictive function
- Decision boundary
- Loss function
- Gradient descent
- Multi-class classification
- Conclusion

Disease class prediction

- Supervised classification problem: Predict the disease class (discrete value) of patient given existing medical data features (tumor size).



More examples

- Examples of binary classification tasks:
  - Email: Spam (1) or not spam (0)
  - Online financial transaction: Fraudulent (1) or legitimate (0)

$y = \{0, 1\}$ Binary variable ≠ Continuous variable (regression)

$y \in \mathbb{R}$

Benign Malign

- From binary to multi-class classification:
  - Email: Spam (0), work (1), friends (2), family (3)
  - Medical diseases: Benign (0), malign I (1), malign II (2), malign III (3)

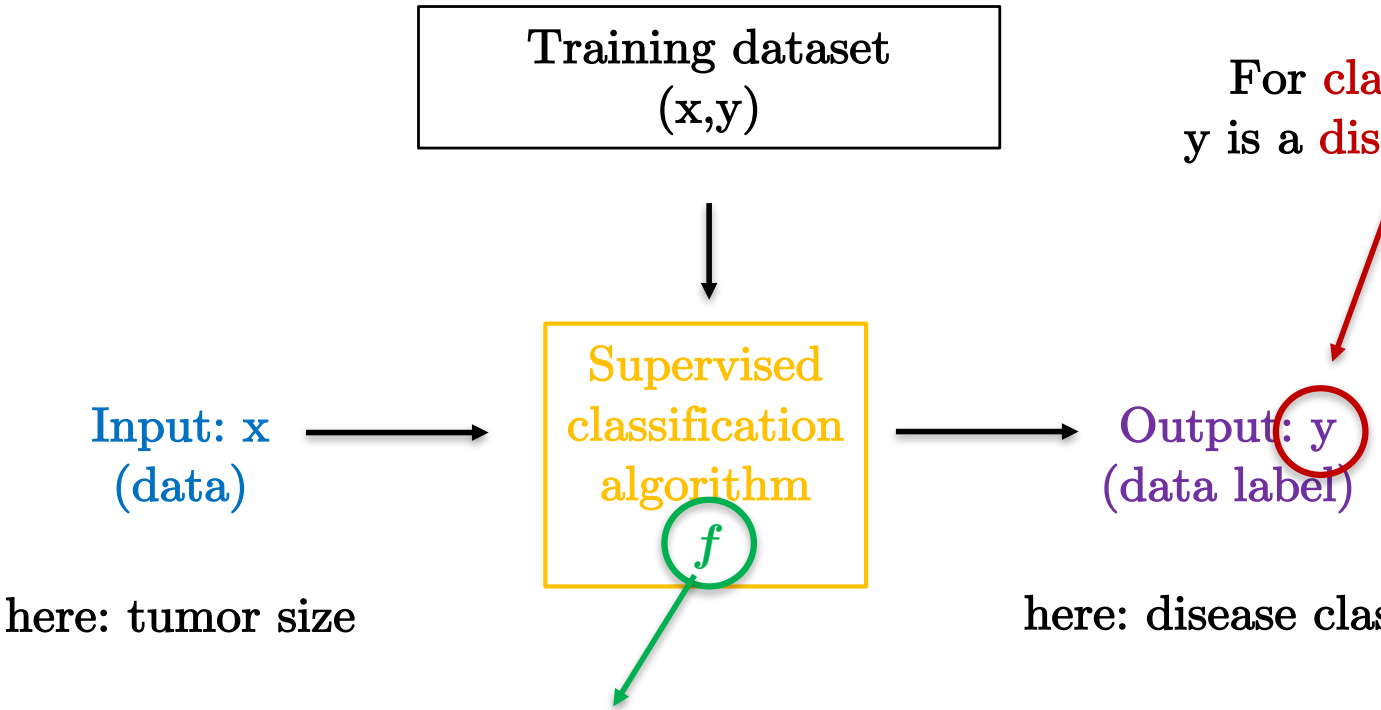
Multi-value variable: $y = \{0, 1, 2, ..., K\}$

Formalization

- Supervised classification learning:

Reminder:
For regression,
y is a scalar

For classification,
y is a discrete variable



Outline

- Classification problem
- **Predictive function**
- Decision boundary
- Loss function
- Gradient descent
- Multi-class classification
- Conclusion

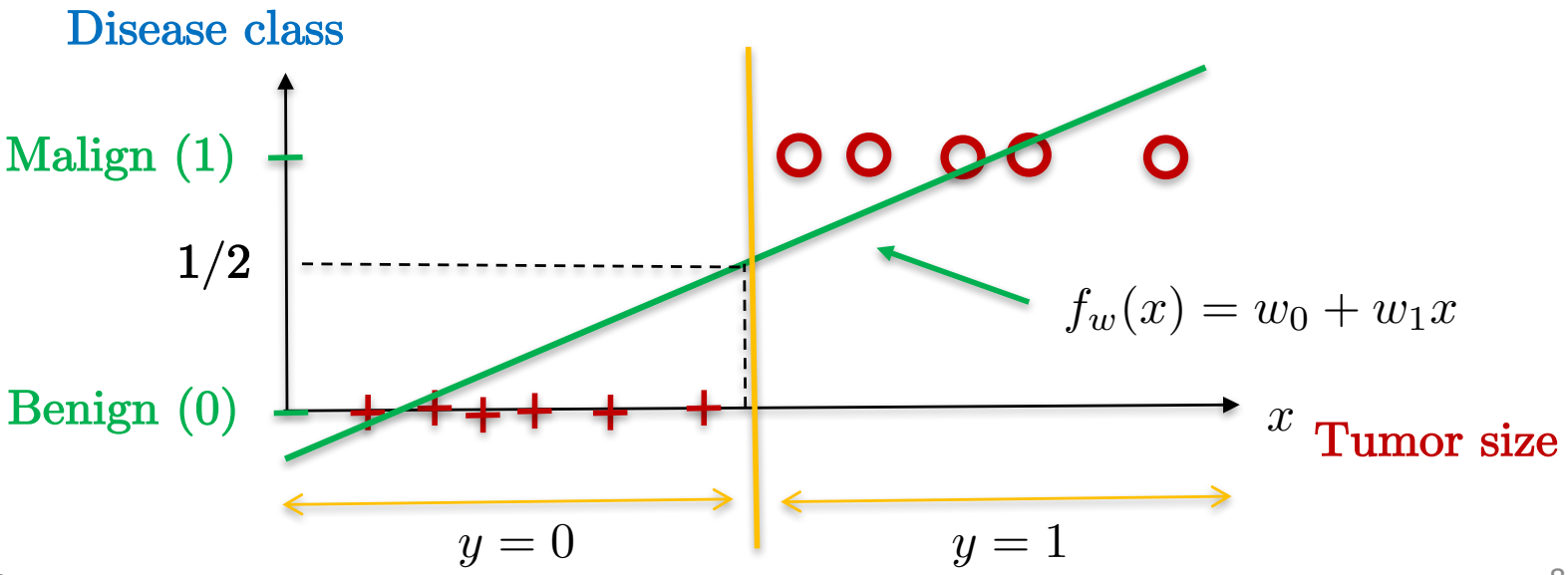
Model representation

- How to represent a (discrete) class prediction function?
  - Linear model? (like for regression)

$$f_w(x) = w_0 + w_1 x$$

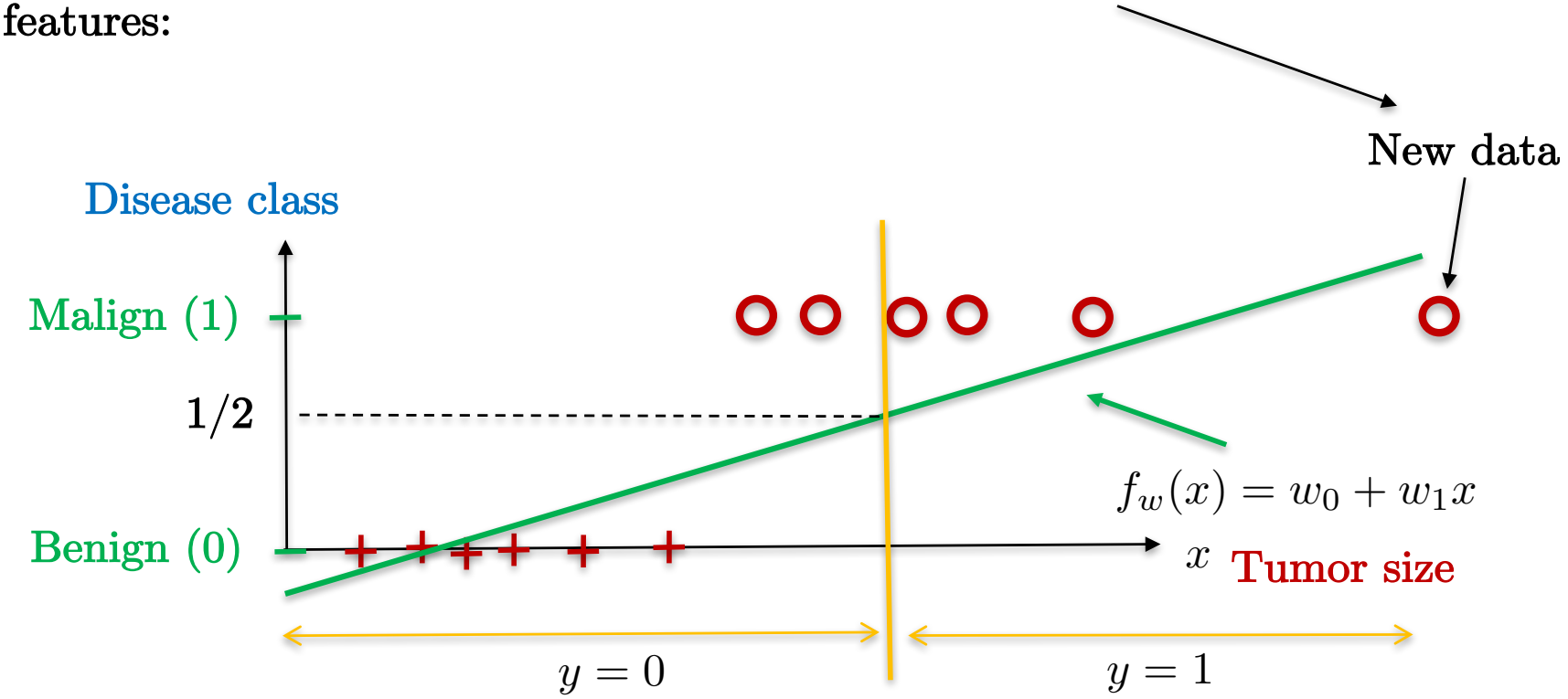
  - Class prediction might be:

$$\text{if } f_w(x) \geq 0.5 \text{ then predict } y = 1$$
$$\text{if } f_w(x) < 0.5 \text{ then predict } y = 0$$



Limitation of linear model

- Linear classification models are not robust to large variations of data features:



- The new data has changed significantly the classification result.

  $\Rightarrow$ Linear model is not a good solution to the classification problem.

Model representation

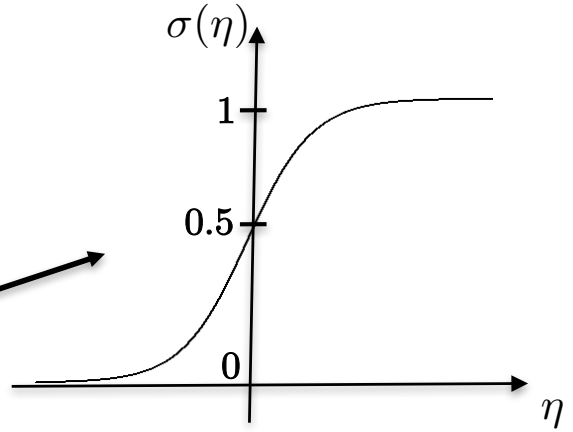
- **Prediction function for classification of *d*-dim data:**

$$\begin{cases} f_w(x) = \sigma(w^T x) \\ \sigma(\eta) = \dfrac{1}{1+e^{-\eta}} \end{cases}$$

Logistic/ sigmoid function

$$f_w(x) = \frac{1}{1+e^{-w^T x}}$$

Logistic regression/ classification function



Sigmoid is like a smooth gate

with $w^T x = w_0 + w_1 x_{(1)} + \ldots + w_d x_{(d)}$

$$x = \begin{bmatrix} 1 \\ x_{(1)} \\ x_{(2)} \\ \vdots \\ x_{(d)} \end{bmatrix} \qquad w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

Probabilistic interpretation

- The prediction function with logistic regression is a probability function:

$$f_w(x) = \Pr_w(y = 1|x)$$

Probability to have y=1 given data x

Probability is parametrized by w

Example: If x = 5mm (tumor size) and $f_w(x) = 0.3$ then the patient has 30% chance of tumor being malign.

- New notation for prediction function:

$$f_w(x) \quad \Rightarrow \quad p_w(x) = \Pr_w(y = 1|x) = \frac{1}{1 + e^{-w^T x}}$$

Probability function

Quiz

- What is the probability to have y=0 given data x?

Use probability property that

$$\Pr_w(y=1|x) + \Pr_w(y=0|x) = 1 \quad \forall x$$

$$\Pr_w(y=0|x) = 1 - \Pr_w(y=1|x) = 1 - p_w(x) = 1 - \frac{1}{1+e^{-w^T x}}$$

Probability to have y=0 given data x

Outline

- Classification problem
- Predictive function
- **Decision boundary**
- Loss function
- Gradient descent
- Multi-class classification
- Conclusion

Class prediction

- Soft (continuous) class predictive function:

$$p_w(x) = \sigma(w^T x) = \frac{1}{1 + e^{-w^T x}}$$

- Observe this predicative function is not a "hard" prediction (discrete value {0,1} to have class 1 or class 2), but a "soft" prediction (probability value between [0,1] to have class 1 or class 2).

- Hard (discrete) class predicative function:

$$\text{if } p_w(x) \geq 0.5 \text{ then } y = 1$$
$$\text{if } p_w(x) < 0.5 \text{ then } y = 0$$

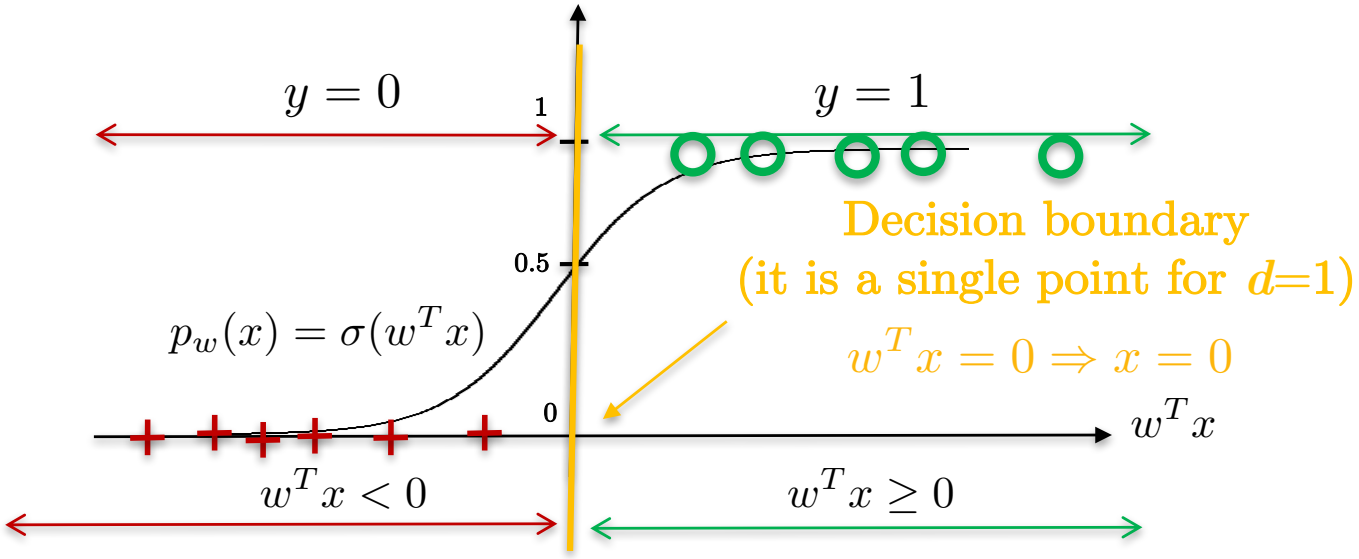
Decision boundary

- Interpretation of $\begin{cases} \text{if } p_w(x) \geq 0.5 \text{ then } y = 1 \\ \text{if } p_w(x) < 0.5 \text{ then } y = 0 \end{cases}$

As $\sigma(\eta = w^T x) \geq 0.5$ when $\eta = w^T x \geq 0$

Therefore $p_w(x) = \sigma(w^T x) \geq 0.5$ if $w^T x \geq 0$ (and $y = 1$)

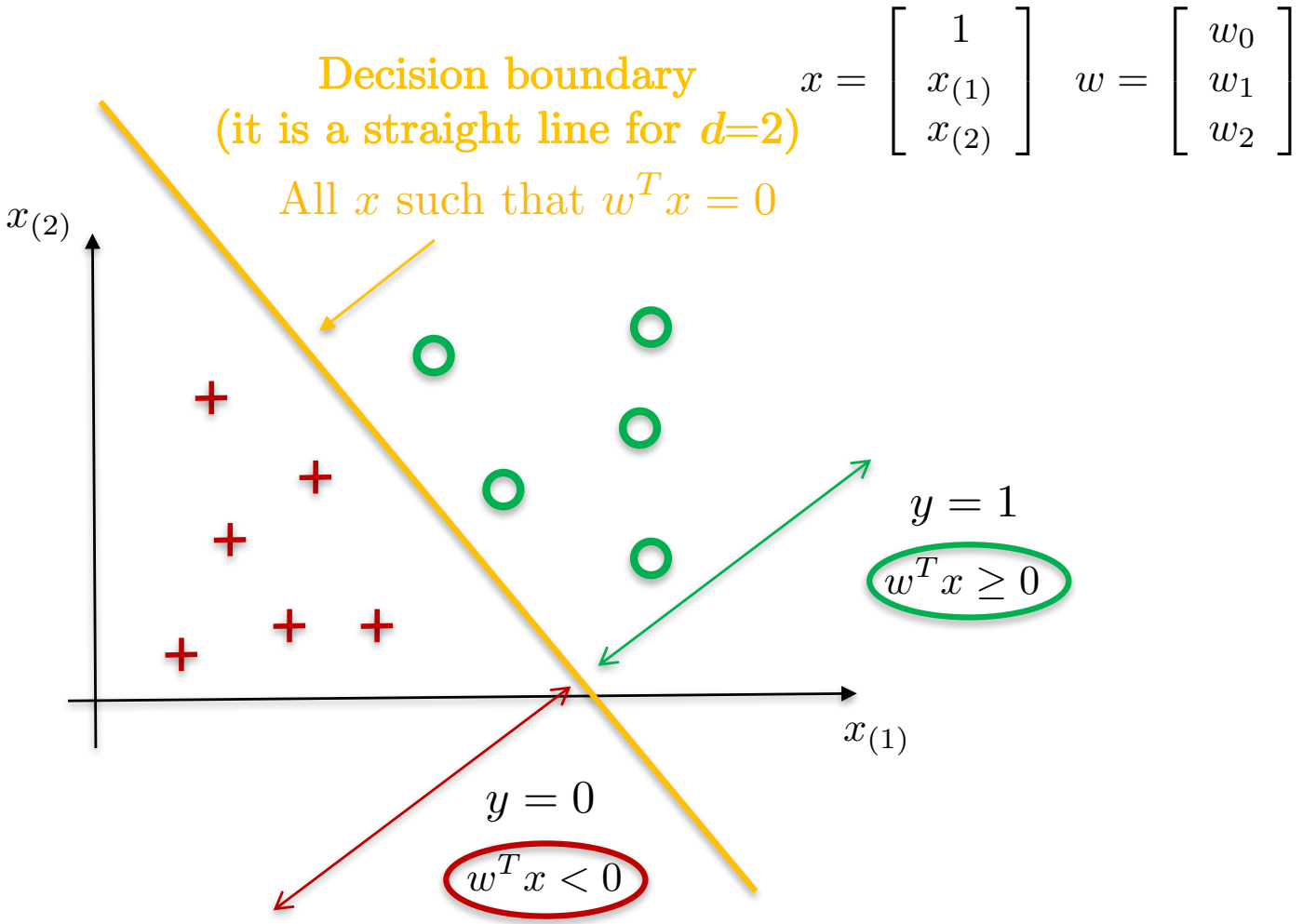
And $p_w(x) = \sigma(w^T x) < 0.5$ if $w^T x < 0$ (and $y = 0$)



Decision boundary for d=2 features

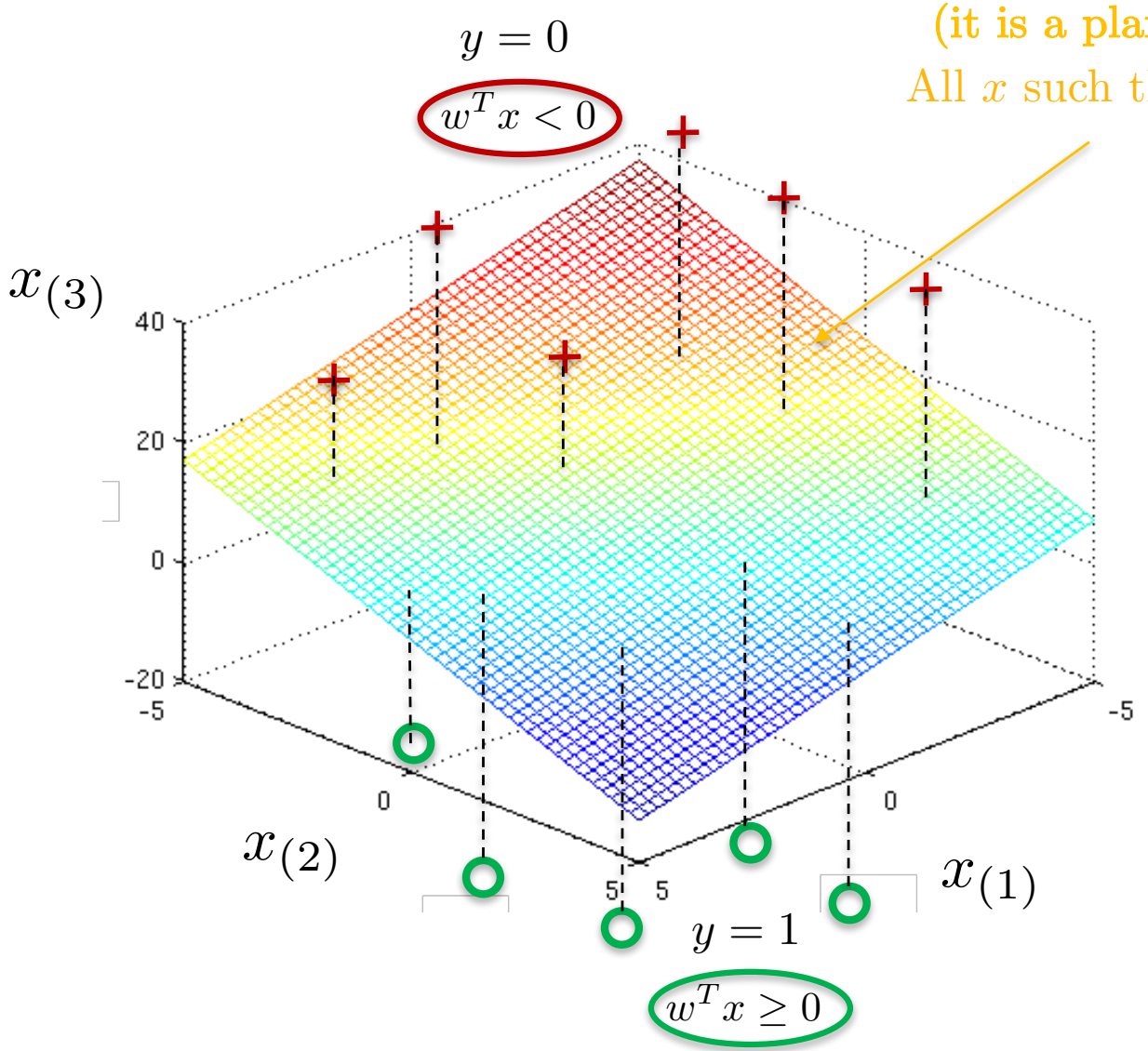
- Decision boundary in higher dimensional spaces:

$$p_w(x) = \sigma(w_0 + w_1 x_{(1)} + w_2 x_{(2)}) = \sigma(w^T x)$$



Decision boundary for d features

- Plan in 3D and hyper-plan in d-D



Non-linear decision boundary

- **Beyond flat boundaries (straight lines, plans):**

$$p_w(x) = \sigma(\underbrace{w_0 + w_1 x_{(1)} + w_2 x_{(2)} + w_3 x_{(1)}^2 + w_4 x_{(2)}^2}_{\text{Quadratic function}}) = \sigma(w^T x)$$

$$w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \qquad x = \begin{bmatrix} 1 \\ x_{(1)} \\ x_{(2)} \\ x_{(1)}^2 \\ x_{(2)}^2 \end{bmatrix}$$

- **Class decision function:**

$$\text{if } p_w(x) \geq 0.5 \text{ or } w^T x \geq 0 \text{ then } y = 1$$
$$\text{if } p_w(x) < 0.5 \text{ or } w^T x < 0 \text{ then } y = 0$$

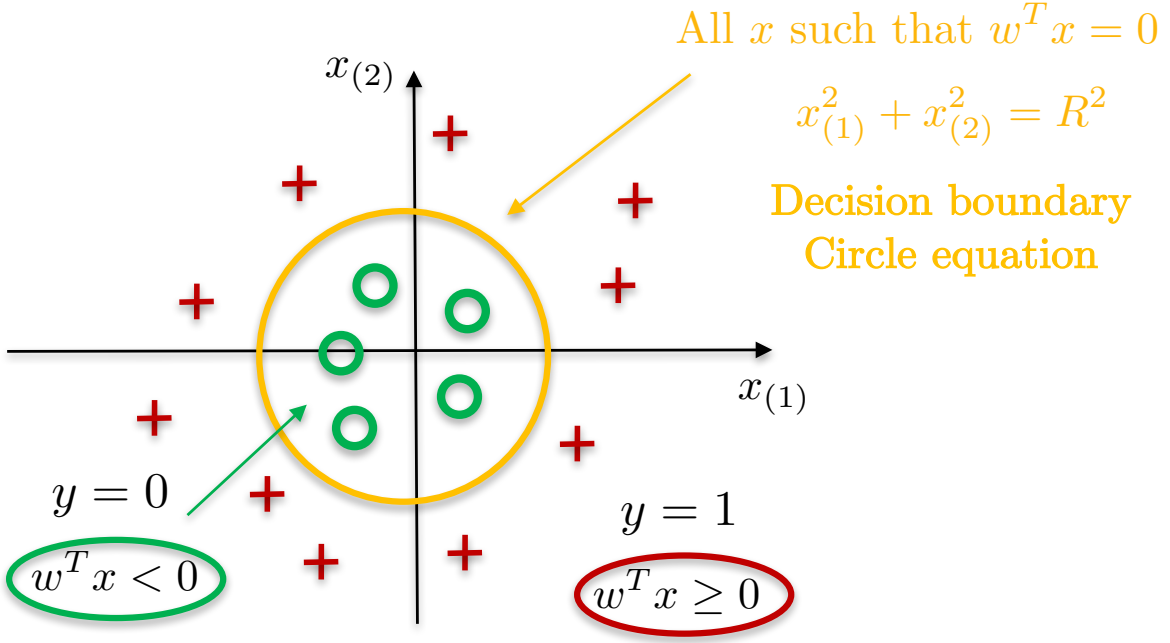
Non-linear decision boundary

- **Example:**

w_0=-R^2, w_1=w_2=0, w_3=w_4=1

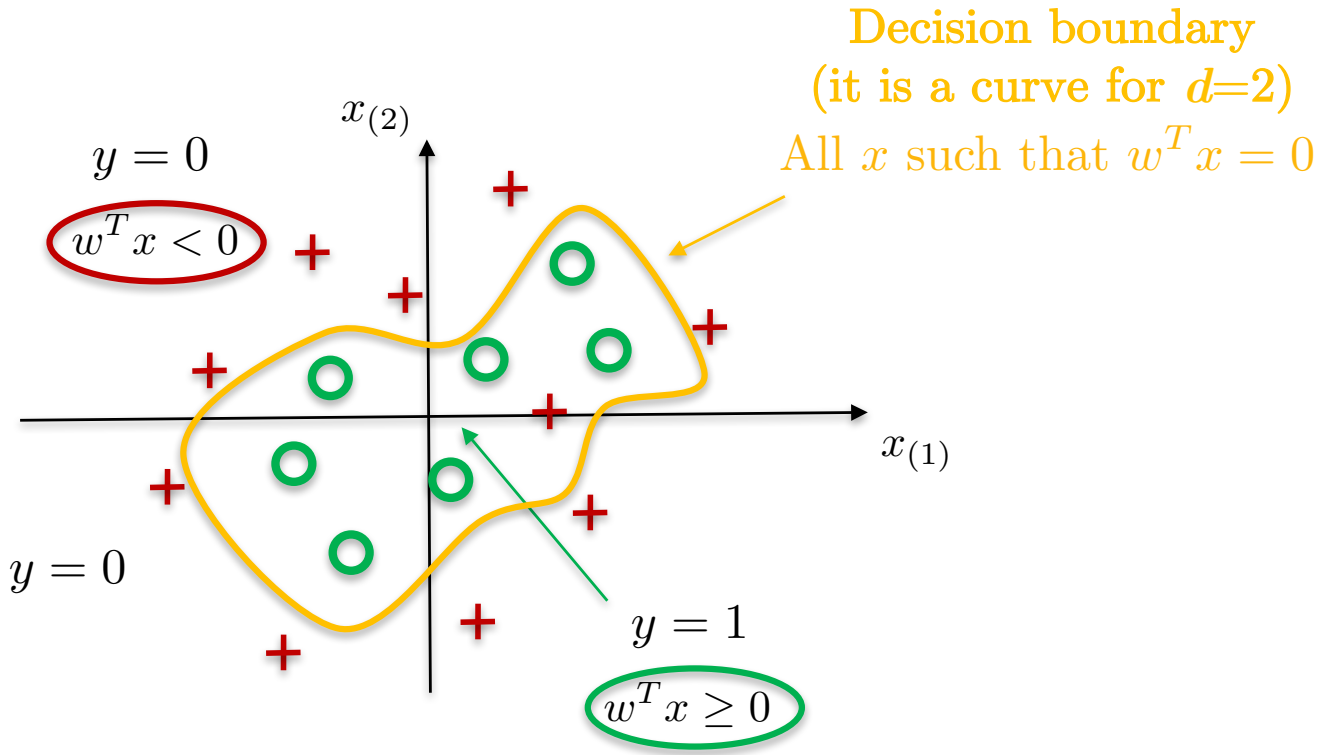
$$\Downarrow$$

$$w^T x = -R^2 + x_{(1)}^2 + x_{(2)}^2$$



Quiz

- What is the most general shape/geometry of a non-linear boundary decision?



Outline

- Classification problem
- Predictive function
- Decision boundary
- **Loss function**
- Gradient descent
- Multi-class classification
- Conclusion

Loss function

- Predictive function:

$$p_w(x) = \sigma(w^T x) = \frac{1}{1 + e^{-w^T x}}$$

- How to choose the parameters w of the predictive function p_w?

  We need:

  - A loss/cost function to assess the prediction.
  - A training set of examples (x_i, y_i) (supervised learning)

- Candidate: Loss function used for regression?

$$L(w) = \frac{1}{n} \sum_{i=1}^{n} \left(p_w(x_i) - y_i \right)^2$$ Mean square error (MSE)

Good choice for classification?

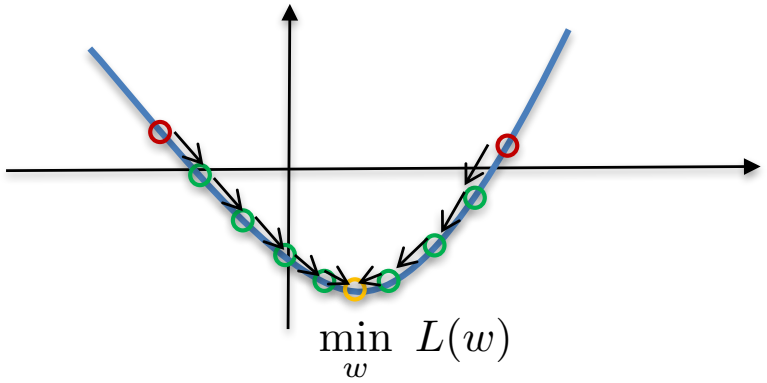
MSE loss for regression and classification

- Linear regression predictive function:

$$f_w(x) = w^T x$$

- MSE loss:

$$L(w) = \frac{1}{n}\sum_{i=1}^{n}\left(w^T x_i - y_i\right)^2$$



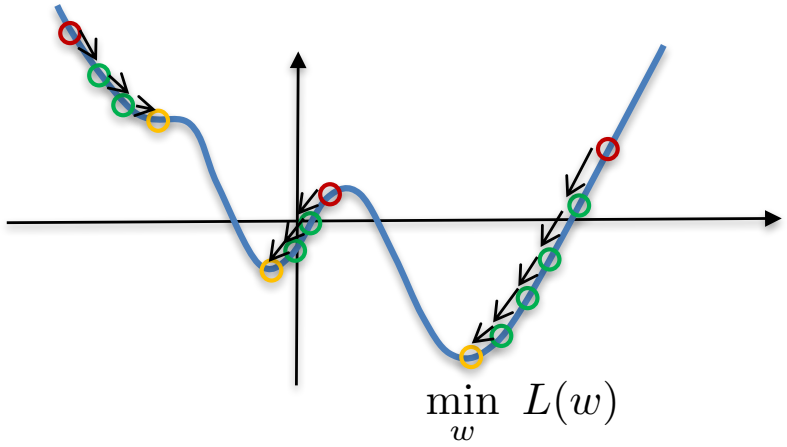
- L function is convex ☺
  - GD guarantees to find (global) minimum

- Classification predictive function:

$$p_w(x) = \frac{1}{1 + e^{-w^T x}}$$

- MSE loss:

$$L(w) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{1 + e^{-w^T x_i}} - y_i\right)^2$$



- L function is non-convex ☹
  - GD no guaranteed to converge to global minimum

Logistic regression loss

- Most popular classification loss is the logistic regression loss.
  - Note: The name "logistic regression" may be confusing as we deal with the classification task (not the regression task).

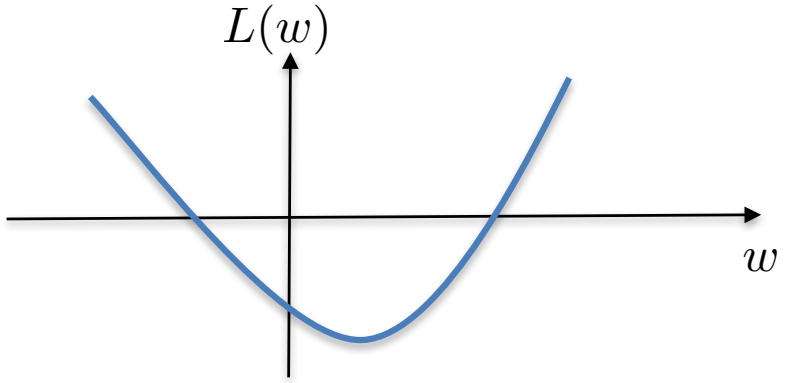
- Definition:

$$L(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(p_w(x_i), y_i)$$

$$\text{with } \ell(p_w(x_i), y_i) = \begin{cases} -\log p_w(x_i) & \text{if} \quad y_i = 1 \\ -\log(1 - p_w(x_i)) & \text{if} \quad y_i = 0 \end{cases}$$

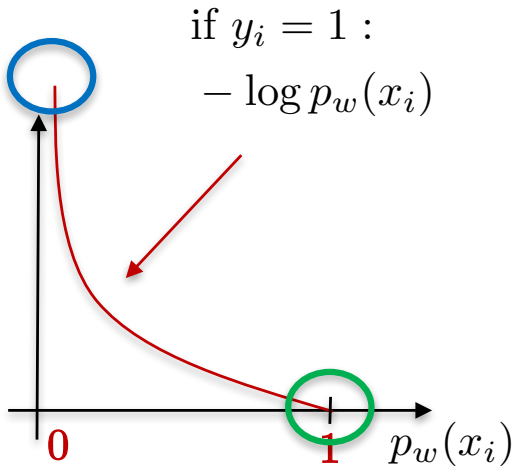
$$\text{and} \quad p_w(x_i) = \frac{1}{1 + e^{-w^T x_i}}$$

- Convexity: Logistic regression function $L(w)$ is convex ☺.

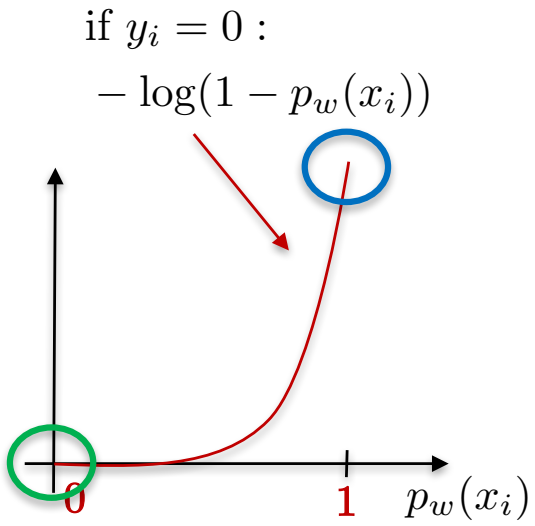


Loss analysis

- Properties of the logistic regression loss:
  - If $y_i=1$ and the predictive function $p_w(x_i)$ predict 1 (correct), we should have:
    $$\ell(p_w(x_i), y_i) = 0$$
  - If $y_i=1$ and the predictive function $p_w(x_i)$ predict 0 (mistake), we should penalize:
    $$\ell(p_w(x_i), y_i) = +\infty$$



- Properties of the logistic regression loss:
  - If $y_i=0$ and the predictive function $p_w(x_i)$ predict 0 (correct), we should have:
    $$\ell(p_w(x_i), y_i) = 0$$
  - If $y_i=0$ and the predictive function $p_w(x_i)$ predict 1 (mistake), we should penalize:
    $$\ell(p_w(x_i), y_i) = +\infty$$



Quiz

- Can we re-write the logistic regression loss is one line? Yes.

$$L(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(p_w(x_i), y_i)$$

$$\text{with } \ell(p_w(x_i), y_i) = \begin{cases} -\log p_w(x_i) & \text{if} \quad y_i = 1 \\ -\log(1 - p_w(x_i)) & \text{if} \quad y_i = 0 \end{cases}$$

$$\Downarrow$$

$$L(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(p_w(x_i), y_i)$$

$$\text{with } \ell(p_w(x_i), y_i) = \underbrace{-y_i \log p_w(x_i)}_{\text{Term=0 if } y_i=0} \underbrace{- (1 - y_i) \log(1 - p_w(x_i))}_{\text{Term=0 if } y_i=1}$$

- One line expression:

$$L(w) = -\frac{1}{n}\sum_{i=1}^{n} \Big(y_i \log p_w(x_i) + (1 - y_i) \log(1 - p_w(x_i)) \Big)$$

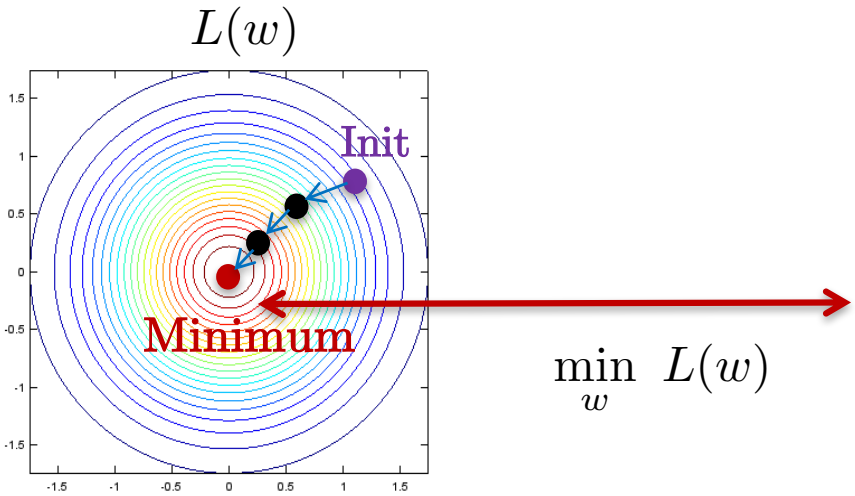
Logistic regression loss

Outline

- Classification problem
- Predictive function
- Decision boundary
- Loss function
- **Gradient descent**
- Multi-class classification
- Conclusion

Gradient descent for logistic regression

- **Prediction function:** $p_w(x) = \frac{1}{1 + e^{-w^T x}}$
- **Parameters:** $w = [w_0, w_1, ..., w_d]$
- **Loss function:** $L(w) = -\frac{1}{n} \sum_{i=1}^{n} \left(y_i \log p_w(x_i) + (1 - y_i) \log(1 - p_w(x_i)) \right)$
- **Optimization:** $\min_w \; L(w)$
- **Gradient descent:** $w_j \leftarrow w_j - \tau \frac{\partial}{\partial w_j} L(w)$



Gradient descent for logistic regression

- **Loss:**

$$L(w) = -\frac{1}{n}\sum_{i=1}^{n} \underbrace{\left(y_i \log p_w(x_i)\right)}_{\text{RHS1}} + \underbrace{(1-y_i)\log(1-p_w(x_i))}_{\text{RHS2}}$$

- **Gradient of RHS1:**

$$\frac{\partial}{\partial w_j}\Big[-\frac{1}{n}\sum_{i=1}^{n} y_i \log p_w(x_i)\Big] = -\frac{1}{n}\sum_{i=1}^{n} y_i \frac{\partial}{\partial w_j}\Big[\log \sigma(w^T x_i)\Big]$$

Chain rule

$$= -\frac{1}{n}\sum_{i=1}^{n} y_i \frac{\sigma'}{\sigma}\frac{\partial}{\partial w_j}\Big[w^T x_i\Big]$$

$$= -\frac{1}{n}\sum_{i=1}^{n} y_i \frac{\sigma(1-\sigma)}{\sigma} x_{i(j)} \qquad \sigma' = \frac{d\sigma}{d\eta} = (1-\sigma(\eta))\sigma(\eta)$$

$$= \frac{1}{n}\sum_{i=1}^{n} y_i(\sigma - 1)x_{i(j)}$$

Chain rule:

$$\frac{\partial}{\partial w}\Big[\underbrace{\log \sigma}_{f}(\underbrace{w^T x}_{z})\Big] = \underbrace{\frac{\partial f}{\partial z}}\underbrace{\frac{\partial z}{\partial w}}$$

$$\frac{\partial \log \sigma(z)}{\partial z} = \frac{\sigma'}{\sigma} \qquad \frac{\partial (w^T x)}{\partial w} = x$$

Gradient descent for logistic regression

- **Loss:**

$$L(w) = -\frac{1}{n}\sum_{i=1}^{n} \underbrace{\Big(y_i \log p_w(x_i)}_{\text{RHS1}} + \underbrace{(1-y_i)\log(1-p_w(x_i))\Big)}_{\text{RHS2}}$$

- **Gradient of RHS2:**

$$\frac{\partial}{\partial w_j}\Big[-\frac{1}{n}\sum_{i=1}^{n}(1-y_i)\log(1-p_w(x_i))\Big] = -\frac{1}{n}\sum_{i=1}^{n}(1-y_i)\frac{\partial}{\partial w_j}\Big[\log(1-\sigma(w^T x_i))\Big]$$

$$= -\frac{1}{n}\sum_{i=1}^{n}(1-y_i)\frac{(1-\sigma)'}{(1-\sigma)}\frac{\partial}{\partial w_j}\Big[w^T x_i\Big]$$

$$= -\frac{1}{n}\sum_{i=1}^{n}(1-y_i)\frac{-\sigma(1-\sigma)}{(1-\sigma)}x_{i(j)}$$

$$= \frac{1}{n}\sum_{i=1}^{n}(1-y_i)\sigma x_{i(j)}$$

Chain rule

$$\frac{\partial}{\partial z}\Big[\log(1-\sigma)\Big] = \frac{(1-\sigma)'}{(1-\sigma)}$$

$$(1-\sigma)' = -\sigma' = -\sigma(1-\sigma(\eta))$$

Gradient descent for logistic regression

- **Loss:**

$$L(w) = -\frac{1}{n}\sum_{i=1}^{n}\Big(y_i \log p_w(x_i) + (1-y_i)\log(1-p_w(x_i))\Big)$$

- **Putting gradients together:**

$$\begin{aligned} w_j &\leftarrow w_j - \tau\frac{\partial}{\partial w_j}L(w) \\ &\leftarrow w_j - \tau\frac{1}{n}\sum_{i=1}^{n}(\sigma(w^T x_i) - y_i)x_{i(j)} \\ &\leftarrow w_j - \tau\frac{1}{n}\sum_{i=1}^{n}(p_w(x_i) - y_i)x_{i(j)} \end{aligned}$$

Quiz

- Have you seen this gradient descent expression before?

  Yes, linear supervised regression.

$$w_j \leftarrow w_j - \tau \frac{1}{n} \sum_{i=1}^{n} (p_w(x_i) - y_i) x_{i(j)}$$

Interestingly, this is the same gradient expression than linear regression. The only difference is the predictive function:

$f_w(x) = w^T x$

Linear regression

$p_w(x) = \frac{1}{1 + e^{-w^T x}}$

Logistic regression/ classifier

$$w_j \leftarrow w_j - \tau \frac{1}{n} \sum_{i=1}^{n} (f_w(x_i) - y_i) x_{i(j)}$$

Outline

- Classification problem
- Predictive function
- Decision boundary
- Loss function
- Gradient descent
- **Multi-class classification**
- Conclusion

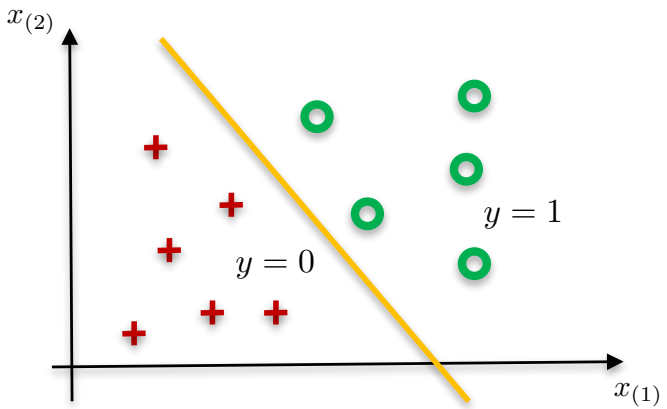
Multi-class problem

- Examples of binary classification tasks:
  - Email: Spam (1) or not spam (0)
  - Online financial transaction: Fraudulent (1) or legitimate (0)

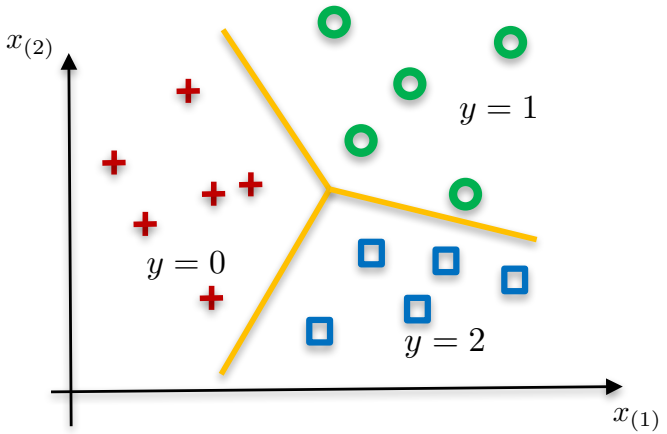
$$y = \{0, 1\} \qquad \text{Binary variable}$$

- From binary to multi-class classification:
  - Email: Spam (0), work (1), friends (2), family (3)
  - Medical diseases: Benign (0), malign I (1), malign II (2), malign III (3)

$$y = \{0, 1, 2, ..., K\} \qquad \text{Multi-value variable}$$



Binary classification

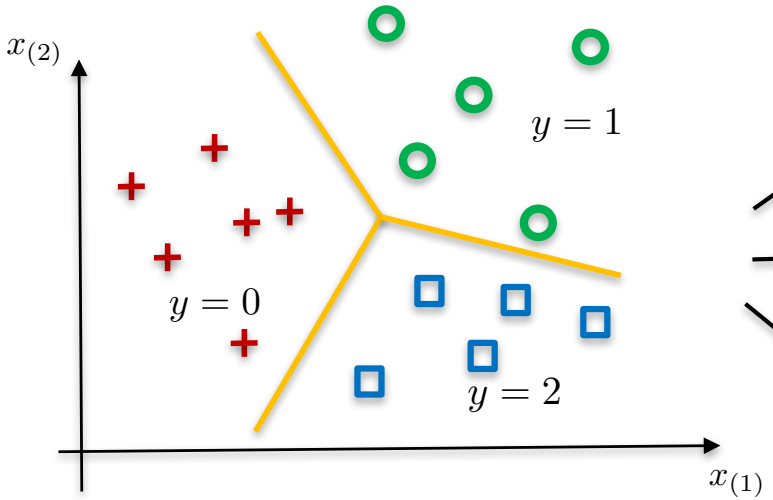


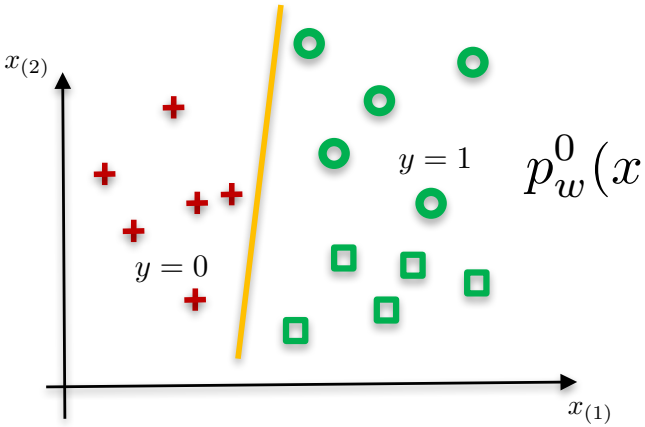
Multi-variate classification

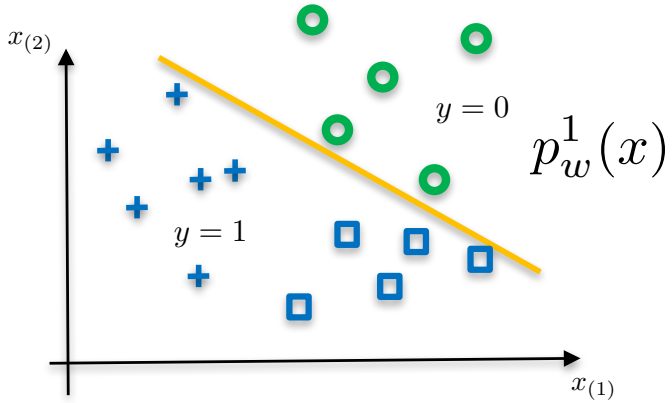
One-vs-all classification problem

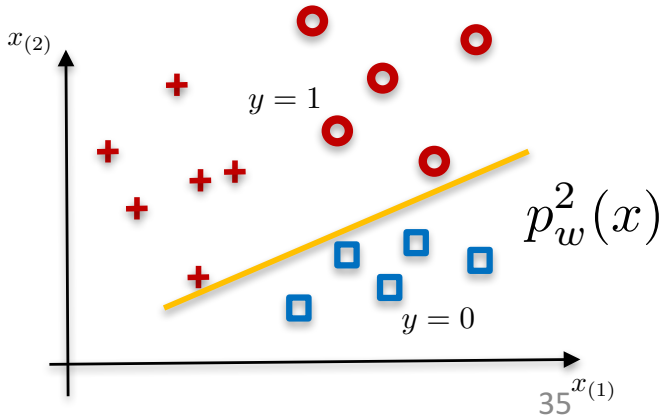
- Two steps:

Step 1 : Learning: Learn K classifiers to recognize of K classes







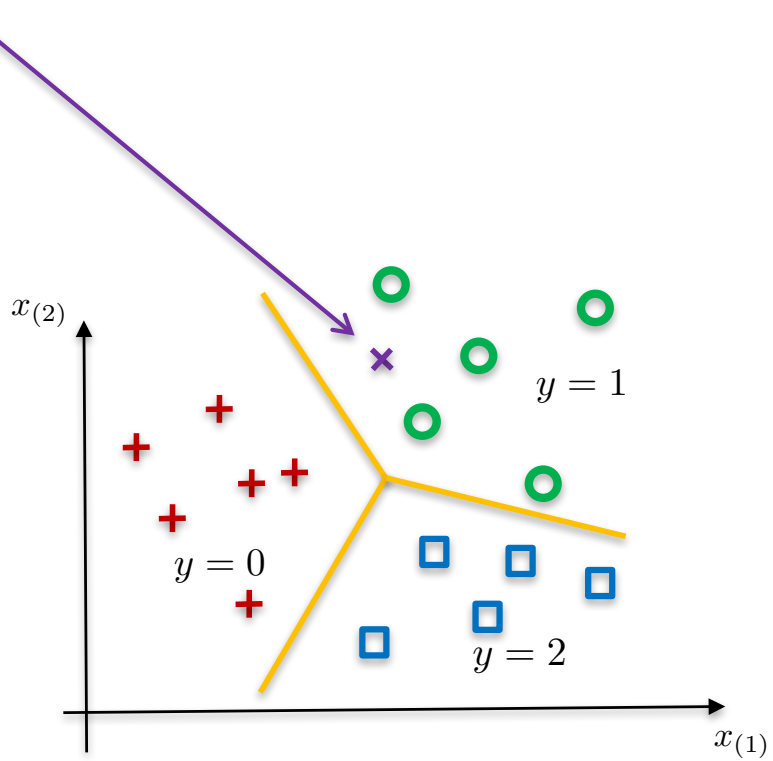


$$p_w^k(x) = \Pr_w(y = i|x) \quad i = 0, 1, 2$$

One-vs-all classification problem

Step 2 : Testing: Classify a new data x with the class k that provides the highest probability:

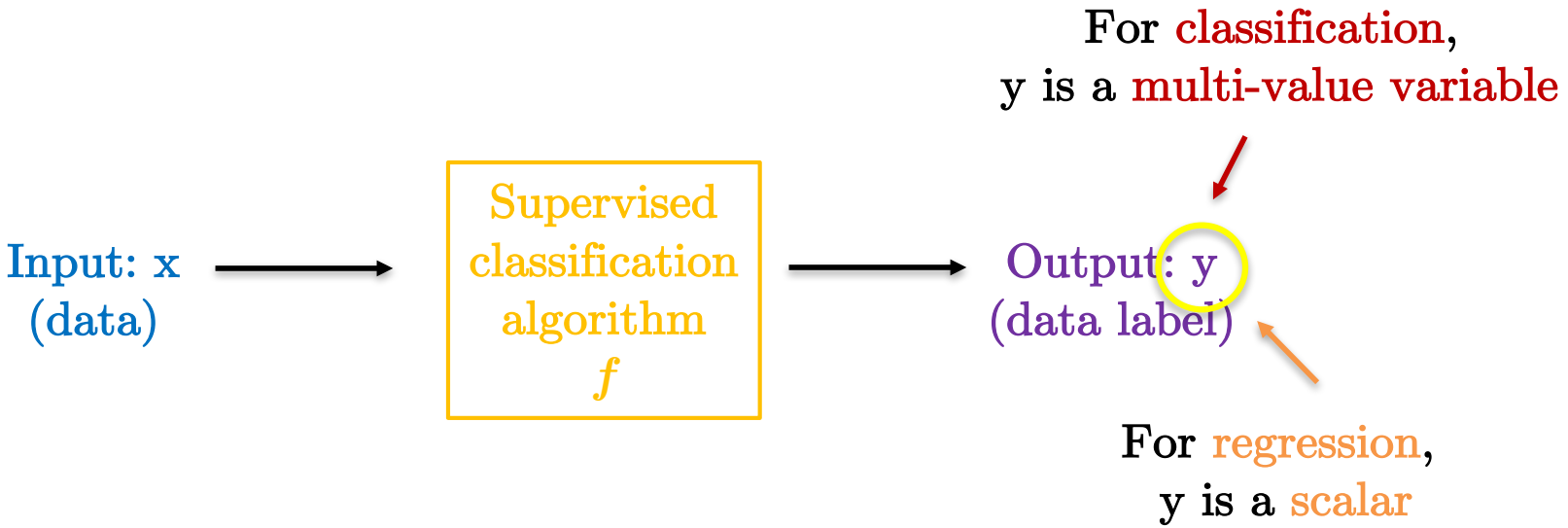
$$k = \arg\max_c p_w^c(x)$$



Quiz

- Why do we want a multi-value variable for classification?

  Why not also using a scalar value like in regression?



- For probabilistic interpretation:

$$p_w^k(x) = \Pr_w(y = i|x) \quad i = 0, 1, 2$$

Outline

- Classification problem
- Predictive function
- Decision boundary
- Loss function
- Gradient descent
- Multi-class classification
- **Conclusion**

Conclusion

- Classification and regression are the two most fundamental tasks in machine learning and data science (many problems can be reduced to solve these tasks).
- Supervised classification is similar to supervised regression:
  - Require training data.
  - Parameter learning can be carried out the same way (optimization).
  - Predictive and loss functions are different.
- Logistic regression loss (classification loss) is the most popular for linear models, and also non-linear models (neural networks).

Coding exercise

- [tutorial04.ipynb]

8. Plot the decision boundary

It is defined by all points

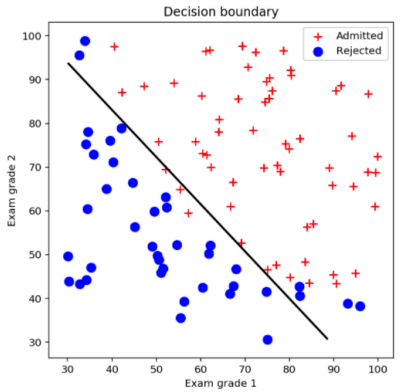
$$x = (x_{(1)}, x_{(2)}) \quad \text{such that} \quad p_w(x) = 0.5$$

Hint: You may use numpy and matplotlib functions `np.meshgrid`, `np.linspace`, `reshape`, `contour`.

In [9]:
```
# compute values p(x) for multiple data points x
x1_min, x1_max = X[:,1].min(), X[:,1].max() # min and max of grade 1
x2_min, x2_max = X[:,2].min(), X[:,2].max() # min and max of grade 2
xx1, xx2 = np.meshgrid(np.linspace(x1_min, x1_max), np.linspace(x2_min, x2_max)) # create meshgrid
X2 = np.ones([np.prod(xx1.shape),3])
X2[:,1] = xx1.reshape(-1)
X2[:,2] = xx2.reshape(-1)
p = f_pred(X2,w)
p = p.reshape(xx1.shape)


# plot
plt.figure(4,figsize=(6,6))
plt.scatter(x1[idx_admit], x2[idx_admit], s=60, c='r', marker='+', linewidths=2, label='Admitted') #YOUR CODE HERE
plt.scatter(x1[idx_rejec], x2[idx_rejec], s=60, c='b', marker='o', linewidths=2, label='Rejected') #YOUR CODE HERE
plt.contour(xx1, xx2, p, [0.5], linewidths=2, colors='k') #YOUR CODE HERE
plt.xlabel('Exam grade 1')
plt.ylabel('Exam grade 2')
plt.legend()
plt.title('Decision boundary')
plt.show()


# record p values
p_gd = p
```

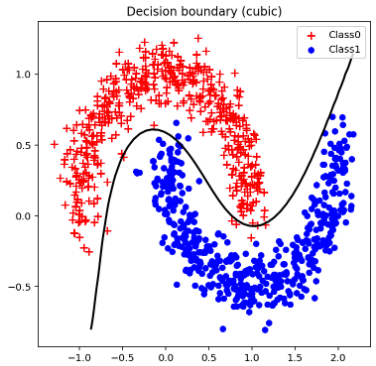


Coding exercise

- tutorial05.ipynb

In [13]:
```
# compute values p(x) for multiple data points x
x1_min, x1_max = X[:,1].min(), X[:,1].max() # min and max of grade 1
x2_min, x2_max = X[:,2].min(), X[:,2].max() # min and max of grade 2
xx1, xx2 = np.meshgrid(np.linspace(x1_min, x1_max), np.linspace(x2_min, x2_max)) # create r
X2 = np.ones([np.prod(xx1.shape),10])
X2[:,1] = xx1.reshape(-1)
X2[:,2] = xx2.reshape(-1)
X2[:,3] = xx1.reshape(-1)**2
X2[:,4] = xx2.reshape(-1)**2
X2[:,5] = xx1.reshape(-1)*xx2.reshape(-1)
X2[:,6] = xx1.reshape(-1)**3
X2[:,7] = xx2.reshape(-1)**3
X2[:,8] = (xx1.reshape(-1)**2)*xx2.reshape(-1)
X2[:,9] = xx1.reshape(-1)*(xx2.reshape(-1)**2)
p = f_pred(X2,w)
p = p.reshape(xx1.shape)


# plot
plt.figure(4,figsize=(6,6))
plt.scatter(x1[idx_class0], x2[idx_class0], s=60, c='r', marker='+', label='Class0')
plt.scatter(x1[idx_class1], x2[idx_class1], s=30, c='b', marker='o', label='Class1')
plt.contour(xx1, xx2, p, [0.5], linewidths=2, colors='k')
plt.legend()
plt.title('Decision boundary (cubic)')
plt.show()
```



Questions?