Day 4: Classification

Summer STEM: Machine Learning

Department of Electrical and Computer Engineering
NYU Tandon School of Engineering
Brooklyn, New York

August 6, 2020

Outline

1 Review

2 Logistic Regression

3 Lab: Diagnosing Breast Cancer

4 Multiclass Classificaiton

5 Lab: Iris Dataset

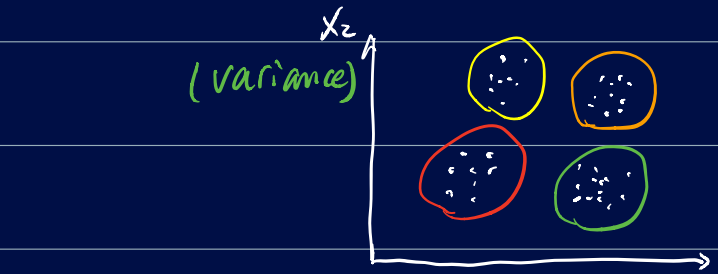
Machine Learning { unsupervised
Supervised (regression / classification)

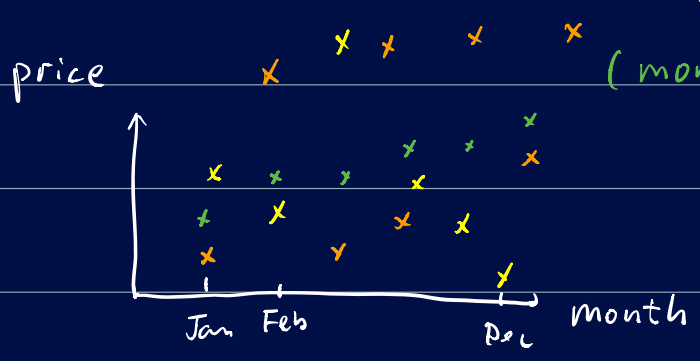
- learn patterns / trends from data
- without specific rules / instructions (autonomous)
- predict

Data features, labels

Supervised data with labels

unsupervised learning data without labels

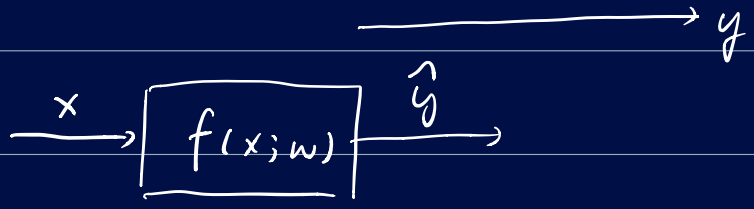




Supervised Learning

- Classification labels: different classes
  discrete number
  binary $\{0, 1\}$
  ↑ cats ↑ dogs
  $\{0, 1, 2\}$ multi-class classification
  ↑ planes ↑ trains ↑ cars

- Regression labels: real numbers



w: $f(x; w) = w_0 + w_1 x$ $w = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$

Vector-valued features $x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix}$

Remark: We use the subscript i in x_i to denote the i-th sample; but sometimes we use the same notation to denote the i-th component of a vector $\mathbf{x}$.

$$f(x_i; w) = w_0 + w_1 x_{i1} + w_2 x_{i2}, \quad w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

a model parameterized by w

$f(x)$

Scalar-valued feature $\times$ Polynomial with an order of M

$$a_0 x^0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_M x^M$$

the order of this polynomial is the highest power $a_i x^i$ where $a_i \neq 0$

Ex: $x + x^4$ → 4-th order

1 0-th order

$x + 3x^4 + 5x^M$ M-th order if $M \geq 4$

$$f(x; w) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M \quad (w_M \neq 0)$$

$$w = [\, w_0 \; w_1 \; \ldots \; w_M \,]^T$$

Matrix-vector representation of the model

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

$$= \begin{bmatrix} 1 & x \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

let's do this for $i = 1, 2, \ldots N$

$$\hat{y}_1 = \begin{bmatrix} 1 & x_1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

$$\hat{y}_2 = \begin{bmatrix} 1 & x_2 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \Rightarrow \hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & x_1 \\ 1 & y_2 \\ \vdots & \\ 1 & x_N \end{bmatrix}}_{X} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

$$\hat{y}_N = \begin{bmatrix} 1 & x_N \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

Vector-valued features

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix} \qquad f(x; w) = w_0 + w_1 x_{i1} + w_2 x_{i2}$$

$$= \begin{bmatrix} 1 & x_{i1} & x_{i2} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & & \\ 1 & x_{N1} & x_{N2} \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}}_{w}$$

The shape of X, w?

X $(N, 3)$

w $(3, 1)$

$\hat{y}$ $(N, 1)$

$$\hat{y} = Xw$$

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ & \vdots & \\ 1 & x_{N1} & x_{N2} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} w_0 + w_1 x_{11} + w_2 x_{12} \\ w_0 + w_1 x_{21} + w_2 x_{22} \\ \\ w_0 + w_1 x_{N1} + w_2 x_{N2} \end{bmatrix}$$

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix} \qquad f(x) = w_0 + w_1 x_{i1} + w_2 x_{i2} = \hat{y}_i$$

$$\hat{y} = Xw \qquad \underline{w = (X^T X)^{-1} X^T y} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

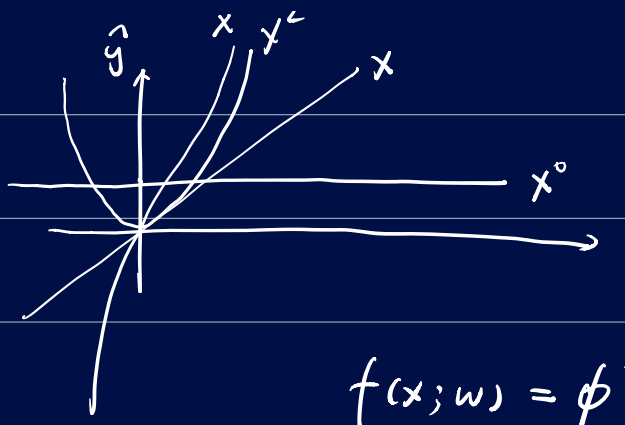
Scalar-valued features x_i

$$f(x_i; w) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M$$

$$X = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^M \\ 1 & x_2 & x_2^2 & \cdots & x_2^M \\ & & \vdots & & \\ 1 & x_N & x_N^2 & \cdots & x_N^M \end{bmatrix} \qquad \underline{Xw}$$

x → feature transformation $\phi(x) = [1 \; x \; x^2 \cdots x^M]^T$

$\underline{[1 \; x \; x^2 \cdots x^M]^T}$ $\qquad \phi_j(x) = x^j$ basis function



$$\begin{bmatrix} 1 & x & x^2 & \cdots & x^M \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{bmatrix}$$

$$f(x; w) = \phi^T(x)\, w = w^T \phi(x)$$

$$\left(a^T b = b^T a \quad (a, b \text{ vectors}) \right)$$

$$f(x_i; w) = w^T \phi(x_i)$$

Scalar-valued $\quad \phi(x_i) = [1 \; x_i]^T$ $\qquad M = 1$

$$\longrightarrow \phi(x_i) = [1 \; x_i \; x_i^2 \cdots x_i^M]^T$$

Vector-valued $\quad \phi(x_i) = [1 \; x_{i1} \; x_{i2} \cdots x_{iD}]^T$

$x_i = [x_{i1} \; x_{i2} \cdots x_{iD}]^T$

$$f(x_i; w) = w^T \phi(x_i) \qquad M = 2$$

$x_i = [x_{i1} \; x_{i2}]^T$

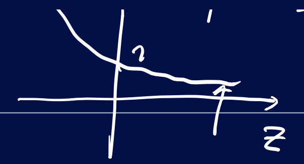
$$f(x_i; w) = w_{00} + w_{10} x_{i1} + w_{01} x_{i2} + w_{20} x_{i1}^2 + w_{02} x_{i2}^2 + w_{11} x_{i1} x_{i2}$$

$$= \sum_j \sum_k w_{jk} x_{i1}^j x_{i2}^k$$

$(j + k \le 2)$

\ * expl-2)

$$\phi_j(x_i) = \exp\left(-\frac{\|x_i - x_j\|^2}{2s^2}\right)$$



Now that we have data, models, how do we evaluate if a model is good or bad?

loss functions $J(w) = \frac{1}{N}\|Y - \hat{Y}\|^2$ (MSE)

$$= \frac{1}{N}\sum_{i=1}^{N}\|y_i - \hat{y}_i\|^2$$

For linear models $\hat{y} = Xw$

$$X = \begin{bmatrix} \phi^T(x_1) \\ \phi^T(x_2) \\ \vdots \\ \phi^T(x_N) \end{bmatrix}$$

(MAE) $J(w) = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$

MAE of 0.7 lbs

MSE of 0.3 lbs^2 $RMSE = \sqrt{MSE}$

$w = (X^TX)^{-1}X^TY$

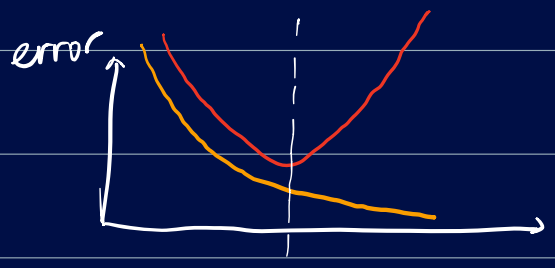
Overfitting and Generalization

- Learn patterns/trends from the data to predict the labels for unseen features
- We say a model generalizes well if it has a good performance for unseen data.
- if a model performs well on the training set but poorly on unseen data ⟶ overfitting

→ How do we alleviate overfitting?

- Reduce M (Polynomials)
- weight-based regularization

} ⇒ limit the complexity of the model



training error

test/validation error

- Hyper-parameters
  - M order of the polynomial
  - λ regularization coefficient

$$J(w) = \frac{1}{N}\sum_{i=1}^{N} \| y_i - \underbrace{w^T \phi(x_i)}_{f(x_i; w)} \|^2 + \lambda \|w\|_2^2$$

MSE — regularization term

$$\lambda \|w\|^2 = \lambda \left(\sqrt{w_0^2 + w_1^2 + \dots + w_M^2} \right)^2$$

$$= \lambda \left(w_0^2 + w_1^2 + \dots w_M^2 \right)$$

$$= \lambda \sum_{j=0}^{M} w_j^2$$

Norm-based regularization

L2 - norm $\quad \|w\|_2 = \sqrt{w_0^2 + w_1^2 + \dots + w_M^2}$

L1 - norm $\quad \|w\|_1 = |w_0| + |w_1| + \dots + |w_M|$

$\lambda \|w\|_2^2 \rightarrow$ Ridge regularization

$\lambda \|w\|_1 \rightarrow$ Lasso regularization

- Ridge regularization $\quad \lambda = 0 \quad \lambda \uparrow \quad \|w^*\| \downarrow$

$$w^* = \left(X_{train}^T X_{train} + \lambda I \right)^{-1} X_{train} Y_{train}$$

- Lasso regularization $\quad \lambda \uparrow \rightarrow$ more components of w become 0

How would it be useful?

$$f(x; w) = w_0 + w_0 x + w_0 x^2 + \dots + w_M x^M$$

Ex x_i: age (x_{i1}), height (x_{i2}), eye color (x_{i3})

y_i: weight

$$\hat{y}_i = w_0 + w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3}$$

Lasso $w_3 \to 0$

Outline

1 Review

2 Logistic Regression

3 Lab: Diagnosing Breast Cancer

4 Multiclass Classificaiton

5 Lab: Iris Dataset

Classification Vs. Regression

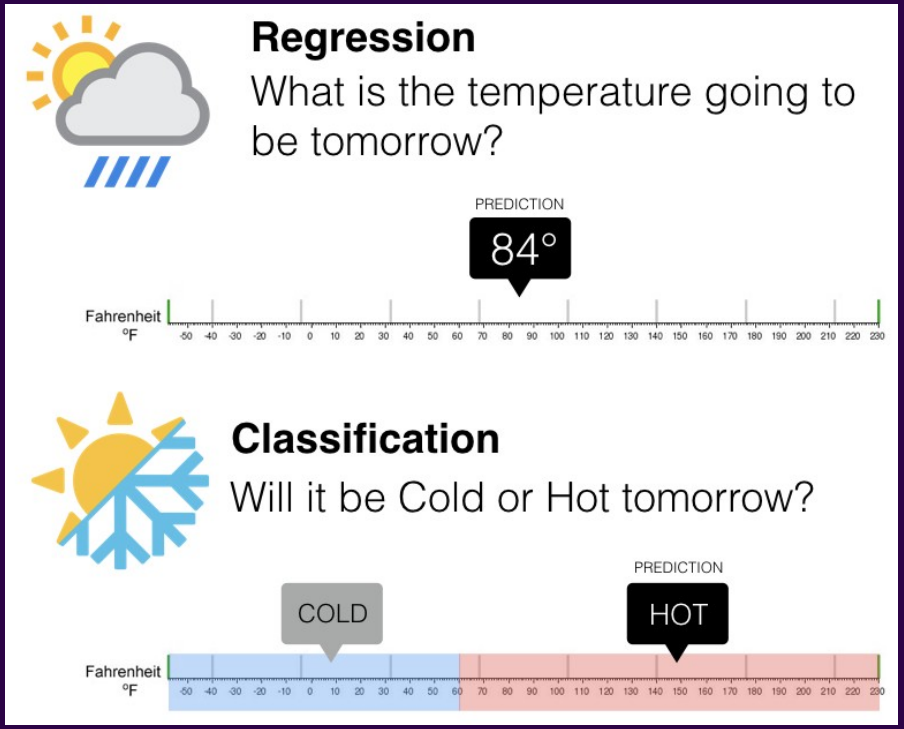
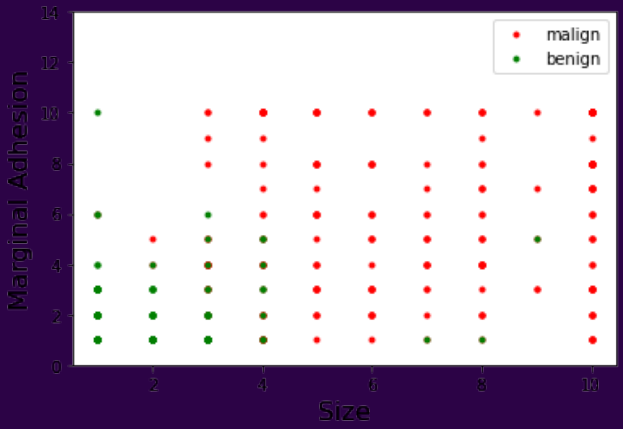
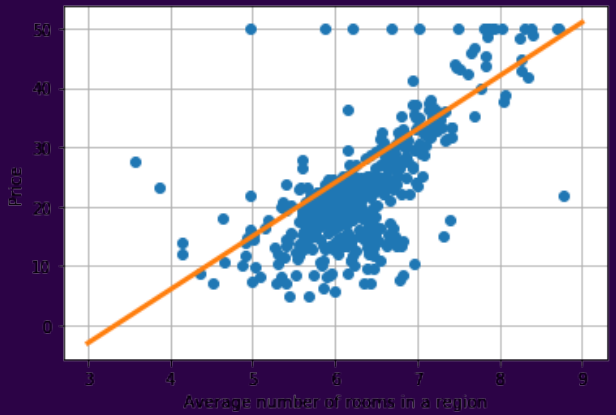


Figure: https://www.pinterest.com/pin/672232681855858622/?lp=true

Classification Vs. Regression



(a) Breast cancer dataset

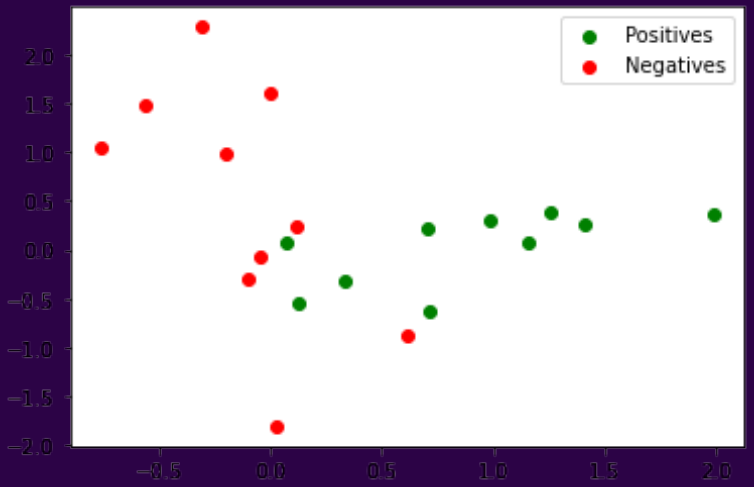


(b) Boston Housing dataset

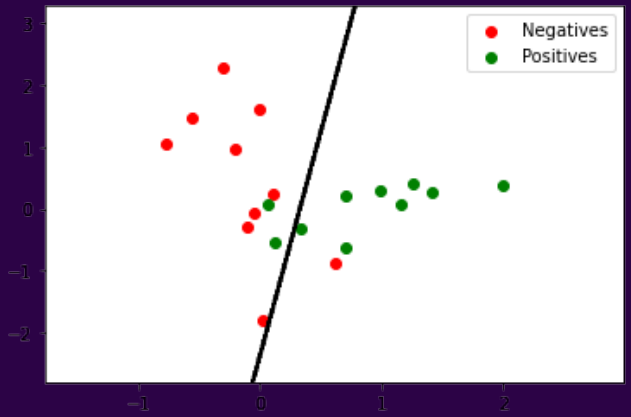
Classification

Given the dataset (x_i, y_i) for $i = 1, 2, \ldots, N$, find a function $f(x)$ (model) so that it can predict the label $\hat{y}$ for some input x, even if it is not in the dataset, i.e. $\hat{y} = f(x)$.

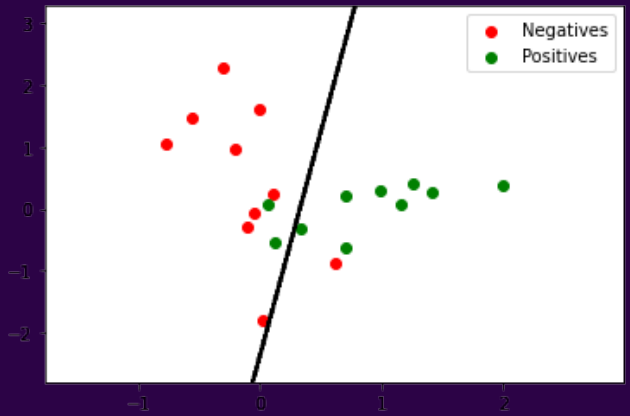
- Positive : $y = 1$
- Negative : $y = 0$



Decision Boundary



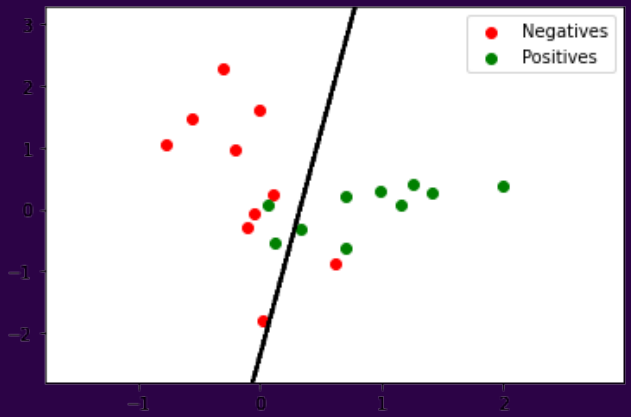
Decision Boundary



- Evaluation metric :

$$\text{Accuracy} = \frac{\text{Number of correct prediction}}{\text{Total number of prediction}}$$

- What is the accuracy in this example ?



- Evaluation metric :

$$\text{Accuracy} = \frac{\text{Number of correct prediction}}{\text{Total number of prediction}} = \frac{17}{20} = 0.85 = 85\%$$

Need for a new model

- What would happen if we used the linear regression model :

$$\hat{y} = w_0 + w_1 x$$

Need for a new model

- What would happen if we used the linear regression model :

$$\hat{y} = w_0 + w_1 x$$

- y is 0 or 1
- $\hat{y}$ will take any value between $-\infty$ and ∞

Need for a new model

- What would happen if we used the linear regression model :

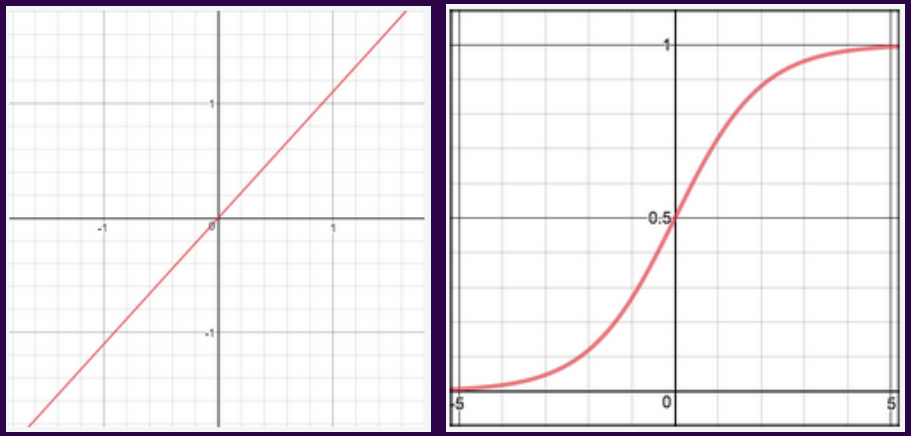
$$\hat{y} = w_0 + w_1 x$$

- y is 0 or 1
- $\hat{y}$ will take any value between $-\infty$ and ∞
- It will be hard to find w_0 and w_1 that make the prediction $\hat{y}$ match the label y.

Sigmoid Function

- By applying the sigmoid function, we enforce $0 \leq \hat{y} \leq 1$

$$\hat{y} = \text{sigmoid}(w_0 + w_1 x) = \frac{1}{1 + e^{-(w_0 + w_1 x)}}$$



(a) Linear model (b) Sigmoid model

A new loss function

- Binary cross entropy loss :

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}\Big[-y_i \log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\Big]$$

pause

- What happens if $y_i = 0$:
$\Big[-y_i \log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\Big] = ?$

A new loss function

- Binary cross entropy loss :

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}\Big[-y_i \log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\Big]$$

- If $y_i = 0$:
$$\Big[-y_i \log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\Big] = -\log(1-\hat{y}_i)$$

A new loss function

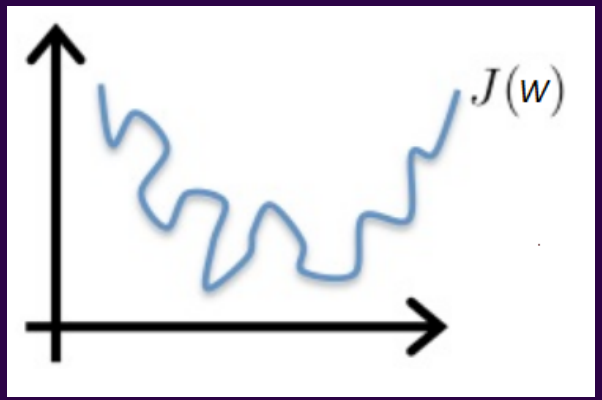
- Binary cross entropy loss :

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}\Big[-y_i\log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\Big]$$

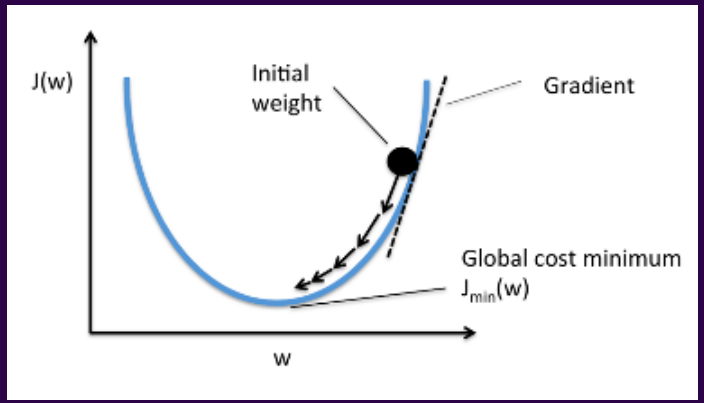
- If $y_i = 0$:
$$\Big[-y_i\log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\Big] = -\log(1-\hat{y}_i)$$
- If $y_i = 1$:
$$\Big[-y_i\log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\Big] = -\log(\hat{y}_i)$$

MSE vs Binary cross entropy loss

- MSE of a logistic function has many local minima.
- The Binary cross entropy loss has only one minimum.



(a) MSE



(b) Binary cross entropy loss

Classifier

$$\hat{y} = \text{sigmoid}(w_0 + w_1 x) = \frac{1}{1 + e^{-(w_0 + w_1 x)}}$$

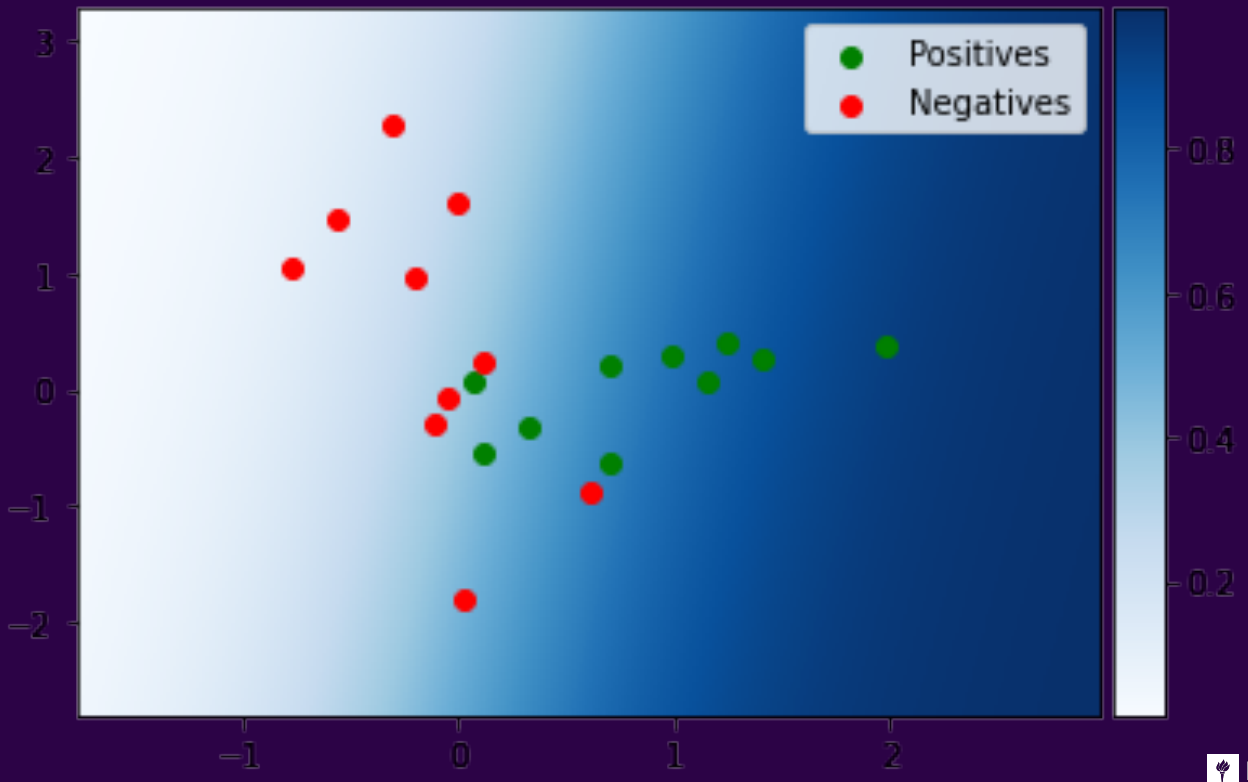
- How to deal with uncertainty ?
  - Thanks to the sigmoid, $\hat{y} = f(x)$ is between 0 and 1.

Classifier

$$\hat{y} = \text{sigmoid}(w_0 + w_1 x) = \frac{1}{1 + e^{-(w_0 + w_1 x)}}$$

- How to deal with uncertainty ?
  - Thanks to the sigmoid, $\hat{y} = f(x)$ is between 0 and 1.
- If $\hat{y}$ is close to 0, the data is probably negative
- If $\hat{y}$ is close to 1, the data is probably positive
- If $\hat{y}$ is around 0.5, we are not sure.

Classifier



Decision Boundary

- Once, we have a classifier outputting a score $0 < \hat{y} < 1$, we need to create a decision rule.

Decision Boundary

- Once, we have a classifier outputting a score $0 < \hat{y} < 1$, we need to create a decision rule.
- Let $0 < t < 1$ be a Threshold :
  - If $\hat{y} > t$, $\hat{y}$ is classified as positive.
  - If $\hat{y} < t$, $\hat{y}$ is classified as negative.

Decision Boundary

- Once, we have a classifier outputting a score $0 < \hat{y} < 1$, we need to create a decision rule.
- Let $0 < t < 1$ be a Threshold :
  - If $\hat{y} > t$, $\hat{y}$ is classified as positive.
  - If $\hat{y} < t$, $\hat{y}$ is classified as negative.
- How to choose t ?

Impact of the threshold

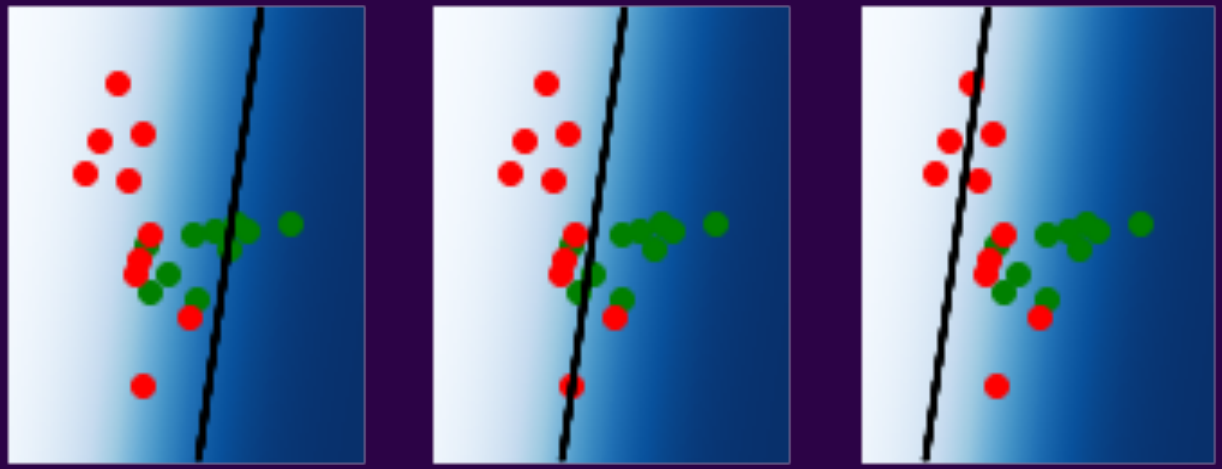


Figure: $t = 0.2, 0.5, 0.8$

Performance metrics for a classifier

- Accuracy of a classifier: percentage of correct classification
- Why accuracy alone is not a good measure for assessing the model ?

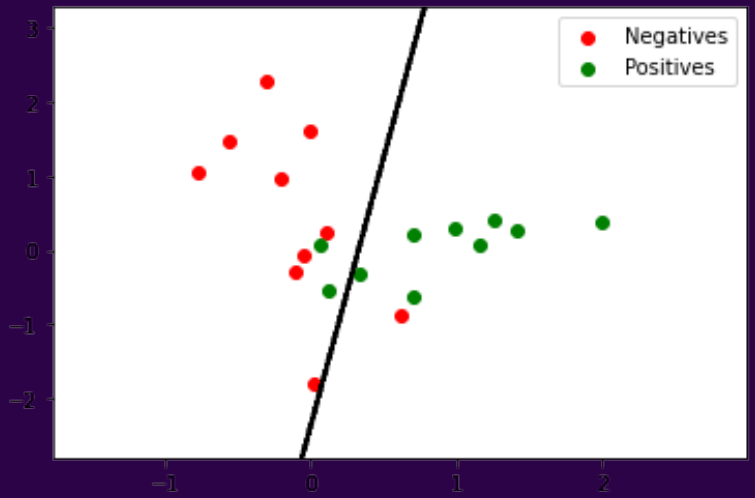
Performance metrics for a classifier

- Accuracy of a classifier: percentage of correct classification
- Why accuracy alone is not a good measure for assessing the model ?
  - Example: A rare disease occurs 1 in ten thousand people
  - A test that classifies everyone as free of the disease can achieve 99.999% accuracy when tested with people drawn randomly from the entire population

Types of Errors in Classification

- Correct predictions:
  - True Positive (TP) : Predict $\hat{y} = 1$ when $y = 1$
  - True Negative (TN) : Predict $\hat{y} = 0$ when $y = 0$
- Two types of errors:
  - False Positive/ False Alarm (FP): $\hat{y} = 1$ when $y = 0$
  - False Negative/ Missed Detection (FN): $\hat{y} = 0$ when $y = 1$

Example



- How many True Positive (TP) are there ?
- How many True Negative (TN) are there ?
- How many False Positive (FP) are there ?
- How many False Negative (FN) are there ?

Other metrics

- Sensitivity/Recall/TPR (How many positives are detected among all positive?)

$$\frac{\text{TP}}{\text{TP} + \text{FN}}$$

- Precision (How many detected positives are actually positive?)

$$\frac{\text{TP}}{\text{TP} + \text{FP}}$$

Linear Models

$f(x) = w_0 + w_1 x$ $\quad x$: scalar

$f(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_D x_D$ $\quad x$: vector $[x_1\ x_2 \ldots x_D]^T$

$f(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M$ $\quad x$: scalar

- Linearity, we say a function $f(x)$ is Linear in x if for any a and x'
  - $f(ax) = af(x)$
  - $f(x + x') = f(x) + f(x')$

Ex: $f(x) = x$ $\quad f(x) = w_1 x$

- $f(ax) = w_1(ax) = a w_1 x = a f(x)$
- $f(x) + f(x') = w_1 x + w_1 x' = w_1(x + x') = f(x + x')$

$f(x) = w_0 + w_1 x$ is not linear in x

$a = 2$ $\quad f(ax) = f(2x) = w_0 + 2w_1 x$

$af(x) = 2f(x) = 2w_0 + 2w_1 x$ $\quad \neq$

$f(x; w) = w^T \phi(x)$ is this linear in w?

$w = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$ if $f(x; aw) = a f(x; w)$

$\phi(x) = \begin{bmatrix} 1 \\ x \end{bmatrix}$ $f(x; w + w') = f(x; w) + f(x; w')$

$f(x; w) = w^T \phi(x)$ $\phi(x)$ $(AB)^T = B^T A^T$

$f(x; aw) = (aw)^T \phi(x) = w^T a^T \phi(x) = w^T a \phi(x) = a w^T \phi(x)$

$= a f(x; w)$ a is a scalar $a^T = a$

$f(x; w+w') = (w+w')^T \phi(x) = (w^T + w'^T)\phi(x)$

$(A+B)^T = A^T + B^T$ $= w^T \phi(x) + w'^T \phi(x) = f(x; w) + f(x; w')$

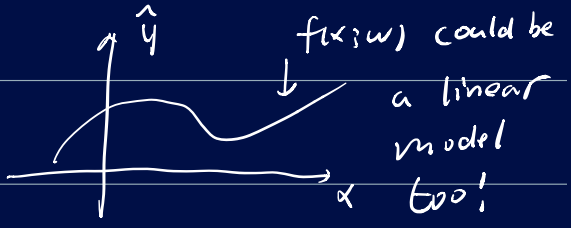
What do we mean by saying the model

$f(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M$ is a linear model? $f(x) = w^T \phi(x)$

$\phi(x) = [1 \; x \; x^2 \ldots x^M]^T$ $f(x) = w^T \phi(x) = w_0 + w_1 x + \cdots + w_M x^M$

A model is a linear model if the function $f(x; w)$ is linear in w (model parameters)

$f(x) = w^T \phi(x)$

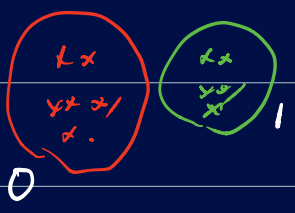


Classification — discrete / finite labels

binary (two-class) classification $\{0, 1\}$

negative positive

	$\hat{y}$	y
TP	1	1
TN	0	0
FP	1	0
FN	0	1



logistic regression

- model $\text{Sigmoid}(w^T \phi(x))$
- loss
- solution ? (w)

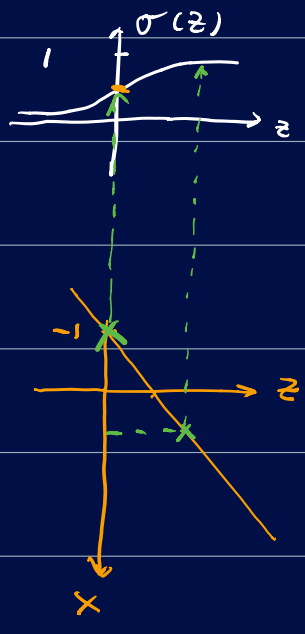
$$\text{Sigmoid}(z) = \frac{1}{1+e^{-z}}$$

$$= \frac{1}{1+1} = 0.5$$

$$\sigma(0) = 0.5$$

$\text{Sigmoid}(x+1) = 0.5$

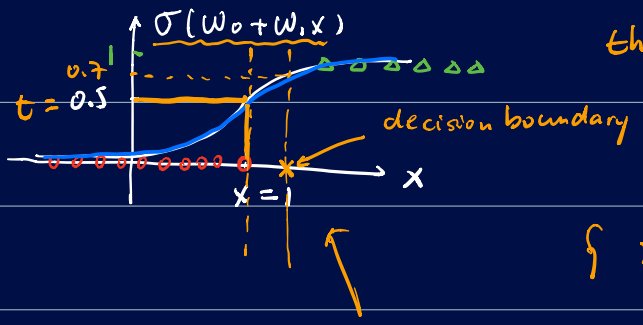
$x \rightarrow ?$



Suppose $\sigma(w^T\phi(x))$ is 0.6

decision rule predicted label $\begin{cases} 1 & \text{if } \sigma(w^T\phi(x)) > 0.7 \\ 0 & \text{if } \sigma(w^T\phi(x)) \leq 0.7 \end{cases}$

threshold $t = 0.5$

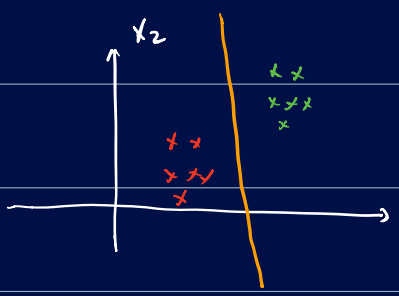


Decision boundary is the set of all feature values x such that $\sigma(w^T\phi(x)) = t$

$\{ x \mid \sigma(w^T\phi(x)) = t \}$

if t is 0.5, what is the decision boundary? {1}

$x = [x_1 \; x_2]^T$

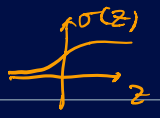


decision boundary is

$\{ x = (x_1, x_2]^T \mid \sigma(w^T\phi(x)) = t \}$

$t = 0.5$ On the decision boundary $w^T\phi(x) = ?$

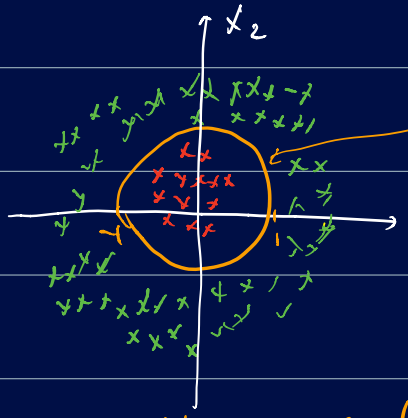
sigmoid $(w^T\phi(x)) = 0.5$ when $w^T\phi(x) = 0$



$0 = w^T\phi(x) = w_0 + w_1x_1 + w_2x_2$

$x_2 = \frac{1}{w_2}(-w_1x_1 - w_0) = -\frac{w_1}{w_2}x_1 - \frac{w_0}{w_2}$

slope



$\{ x \mid \sigma(w^T \phi(x)) = t \}$

We want $x_1^2 + x_2^2 = 1$

It's not possible if $\phi(x) = [1 \;\; x_1 \;\; x_2]^T$

However, if we use $\phi(x) = [1 \;\; x_1^2 \;\; x_2^2]^T$

$$\sigma(w_0 + w_1 x_1^2 + w_2 x_2^2) = t$$

$w = [0 \;\; 1 \;\; 1]^T \quad \sigma(x_1^2 + x_2^2) = t = \sigma(1)$

$$\Rightarrow x_1^2 + x_2^2 = 1$$

logistic regression: $\sigma(w^T \phi(x))$ + decision rule $\begin{matrix} > t \\ \leq t \end{matrix}$

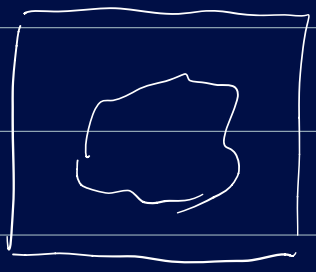
- Loss function: (binary) cross entropy
- How do we find w such that the cross entropy loss is minimized for $\sigma(w^T \phi(x))$?

We don't have a formula for w

sklearn is able to find w — how does it do that?

We can use gradient descent!

Home work



(Size
marginal adhesion)

↓

if it's benign or malignant ?

$x \longrightarrow \hat{y}$

(Size, marg) (benign / malignant)

1 0

.fit (X , y)

$$X = \begin{bmatrix} size_1 & marg_1 \\ size_2 & marg_2 \\ \vdots & \vdots \\ size_N & marg_N \end{bmatrix} \qquad \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

$$X = \begin{bmatrix} \phi^T(x_1) \\ \phi^T(x_2) \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ & \vdots & \\ 1 & x_{N1} & x_{N2} \end{bmatrix}$$

↑ Not needed if fit_intercept = True

```
from sklearn.linear_model import LogisticRegression

regr = LogisticRegression(fit_intercept=True)

regr.fit(X, y)

regr.predict(X)
```

Outline

1 Review

2 Logistic Regression

3 Lab: Diagnosing Breast Cancer

4 Multiclass Classificaiton

5 Lab: Iris Dataset

Lab: Diagnosing Breast Cancer

- We're going to use the breast cancer dataset to predict whether the patients' scans show a malignant tumour or a benign tumour.
- Let's try to find the best linear classifier using logistic regression.

Outline

1 Review

2 Logistic Regression

3 Lab: Diagnosing Breast Cancer

4 Multiclass Classificaiton

5 Lab: Iris Dataset

Multiclass Classificaiton

- Previous model: $f(\mathbf{x}) = \sigma(\mathbf{w}^T \phi(\mathbf{x}))$
- Representing Multiple Classes:
  - One-hot / 1-of-K vectors, ex : 4 Class
  - Class 1 : $\mathbf{y} = [1, 0, 0, 0]$
  - Class 2 : $\mathbf{y} = [0, 1, 0, 0]$
  - Class 3 : $\mathbf{y} = [0, 0, 1, 0]$
  - Class 4 : $\mathbf{y} = [0, 0, 0, 1]$

Multiclass Classificaiton

- Previous model: $f(\mathbf{x}) = \sigma(\mathbf{w}^T \phi(\mathbf{x}))$
- Representing Multiple Classes:
  - One-hot / 1-of-K vectors, ex : 4 Class
  - Class 1 : $\mathbf{y} = [1, 0, 0, 0]$
  - Class 2 : $\mathbf{y} = [0, 1, 0, 0]$
  - Class 3 : $\mathbf{y} = [0, 0, 1, 0]$
  - Class 4 : $\mathbf{y} = [0, 0, 0, 1]$
- Multiple outputs: $f(\mathbf{x}) = \text{softmax}(W^T \phi(\mathbf{x}))$
- Shape of $W^T \phi(\mathbf{x})$: $(K, 1) = (K, D) \times (D, 1)$
- $\text{softmax}(\mathbf{z})_k = \dfrac{e^{z_k}}{\sum_j e^{z_j}}$

Multiclass Classificaiton

- Multiple outputs: $f(\mathbf{x}) = \text{softmax}(\mathbf{z})$ with $\mathbf{z} = W^T \phi(\mathbf{x})$
- $\text{softmax}(\mathbf{z})_k = \dfrac{e^{\mathbf{z}_k}}{\sum_j e^{\mathbf{z}_j}}$
- Softmax example: If $\mathbf{z} = \begin{bmatrix} -1 \\ 2 \\ 1 \\ -4 \end{bmatrix}$ then,

$$\text{softmax}(z) = \begin{bmatrix} \frac{e^{-1}}{e^{-1}+e^{2}+e^{1}+e^{-4}} \\ \frac{e^{2}}{e^{-1}+e^{2}+e^{1}+e^{-4}} \\ \frac{e^{1}}{e^{-1}+e^{2}+e^{1}+e^{-4}} \\ \frac{e^{-4}}{e^{-1}+e^{2}+e^{1}+e^{-4}} \end{bmatrix} \approx \begin{bmatrix} 0.035 \\ 0.704 \\ 0.259 \\ 0.002 \end{bmatrix}$$

Cross-entropy

- Multiple outputs: $\hat{\mathbf{y}}_\mathbf{i} = \text{softmax}(W^T \phi(\mathbf{x}_i))$
- Cross-Entropy: $J(W) = -\sum_{i=1}^{N}\sum_{k=1}^{K} \mathbf{y}_{ik} log(\hat{\mathbf{y}}_{ik})$
- Example : K = 4

$$\text{If, } \mathbf{y}_i = [0, 0, 1, 0] \text{ then, } \sum_{k=1}^{K} \mathbf{y}_{ik} log(\hat{\mathbf{y}}_{ik}) = log(\hat{\mathbf{y}}_{i3})$$

Outline

1 Review

2 Logistic Regression

3 Lab: Diagnosing Breast Cancer

4 Multiclass Classificaiton

5 Lab: Iris Dataset

Lab: Iris Dataset

- Open demo_iris.ipynb