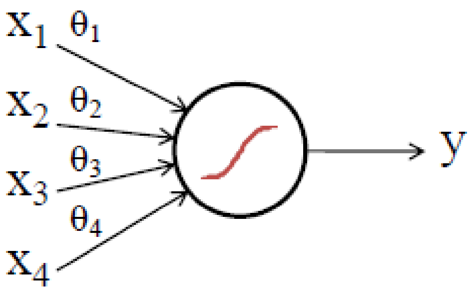
50.007 Machine Learning

Logistic Regression



Berrak Sisman

Assistant Professor, ISTD Pillar, SUTD

Introduction & Content

- **Logistic Regression (week 4)**
- Neural Networks and Deep Learning (week 5)

Instructor: Prof. Berrak Sisman

Email: berrak_sisman@sutd.edu.sg

Feel free to contact me!

What is Logistic Regression?

- A $\boxed{\text{discriminative}}$ **classifier.**
- Logistic regression can be used to classify an observation into one of two classes (like ‘positive sentiment’ and ‘negative sentiment’), or into one of many classes.
- In this lecture, we’ll study **two-class case**.

A **generative model** would have the goal of understanding what dogs look like and what cats look like. You might literally ask such a model to ‘generate’, i.e. draw, a dog.

A **discriminative model**, by contrast, is only trying to learn to distinguish the classes (perhaps without learning much about them)

What is Logistic Regression?

- A discriminative **classifier.**
- Logistic regression can be used to classify an observation into one of two classes (like 'positive sentiment' and 'negative sentiment'), or into one of many classes.
- In this lecture, we'll study **two-class case**.

We'll study generative and discriminative models in week 6! ☺

A **generative model** would have the goal of understanding what dogs look like and what cats look like. You might literally ask such a model to 'generate', i.e. draw, a dog.

A **discriminative model**, by contrast, is only trying to learn to distinguish the classes (perhaps without learning much about them)

What is classification?

- Examples:
  - Email spam classification
  - Classifying online transactions
  - Classifying images (forest, clouds)
  - Tumor: malignant or benign

  ….

$$\boxed{y = \{0,1\} \quad \begin{array}{l}\text{0: “negative class” -> forest images} \\ \text{1: “positive class” -> cloud images}\end{array}}$$

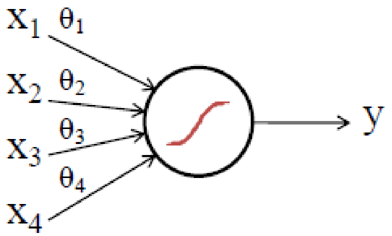
For this representation, we have 2 classes.

Notations for Logistic Regression

- It is a probabilistic classifier that makes use of **supervised** machine learning.
- Machine learning classifiers require a training corpus of M input/output pairs (x^i, y^i).

  We'll use superscripts to refer to individual instances in the training set, for example for sentiment classification each instance might be an individual document to be classified.

- For each input observation x^i, this will be a vector of features $\begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix}$
- We will generally refer to feature i of input x^j as x_i^j or simply x_i.



Big Picture & Motivation

- Support Vector Machines (with me)

- Linear Regression

So where is logistic regression in this picture?

**SVM can perform classification,
why do we need logistic regression?**

Big Picture & Motivation

- Support Vector Machines (with me)

- Linear Regression

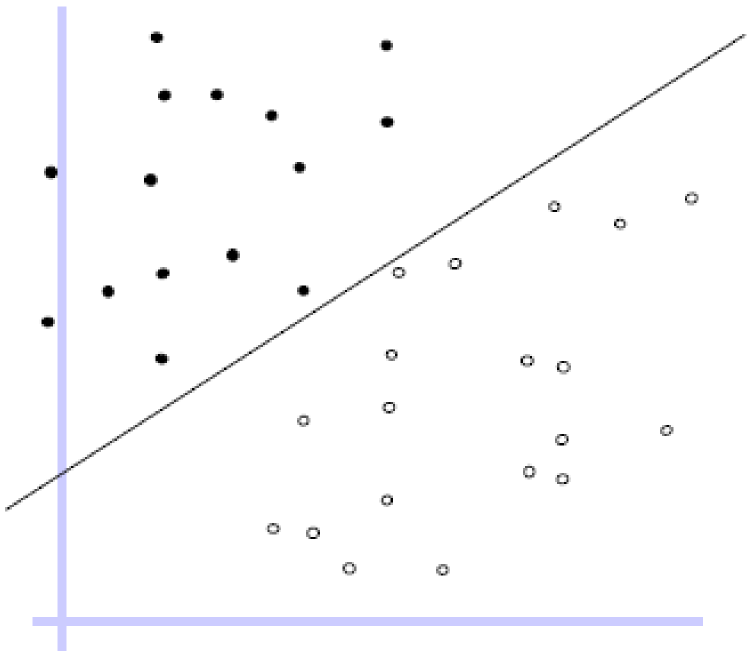
So where is logistic regression in this picture?

**SVM can perform classification,
why do we need logistic regression?**

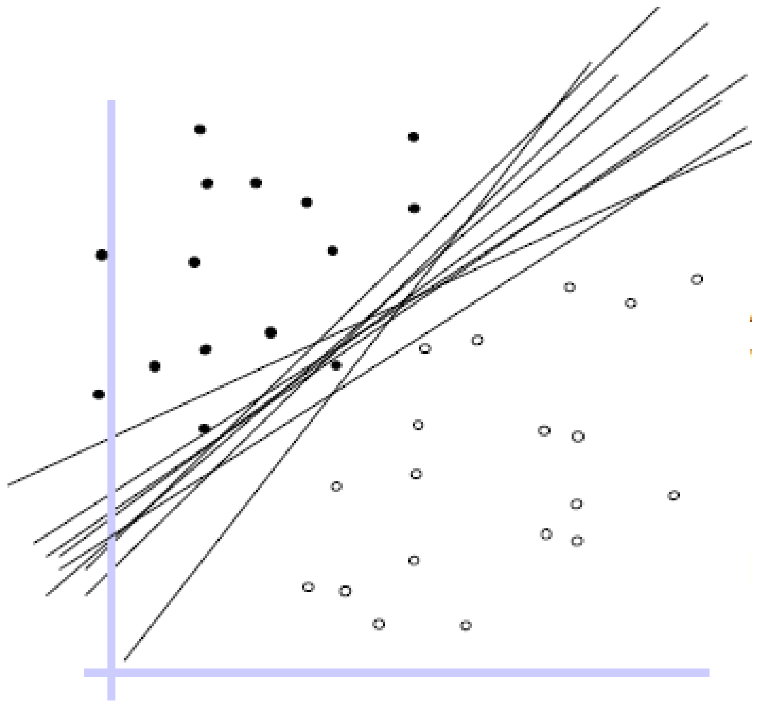
From SVM to Logistic Regression

SVM
Linearly Separable

'Decision boundary'



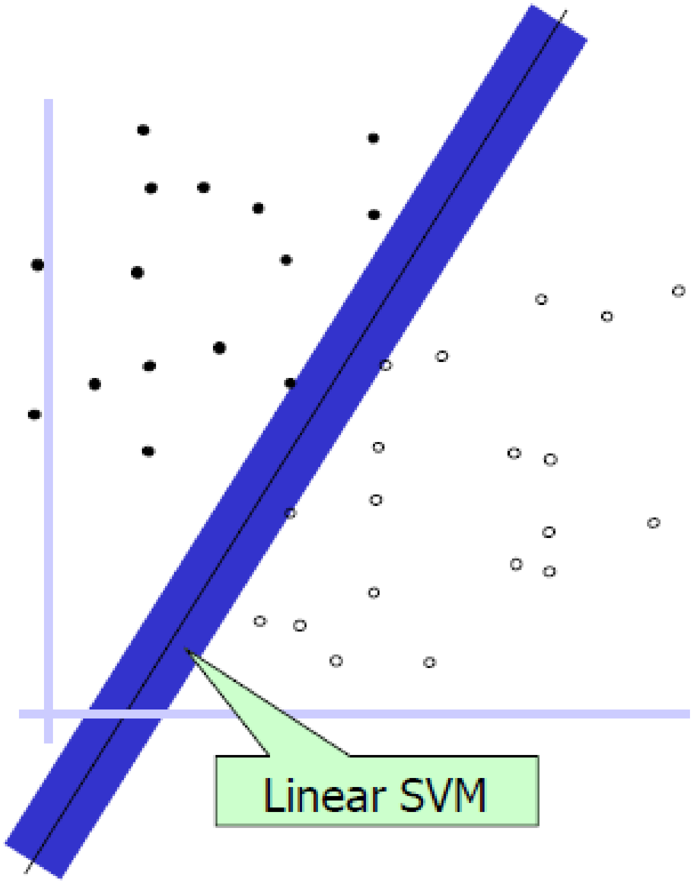
Any of these would be fine…



But which one is the best?

SVM
Linearly Separable

Optimal hyperplane: the one that maximizes the margin



Margin: The width that the boundary can be increased before hitting a data point.

SVM

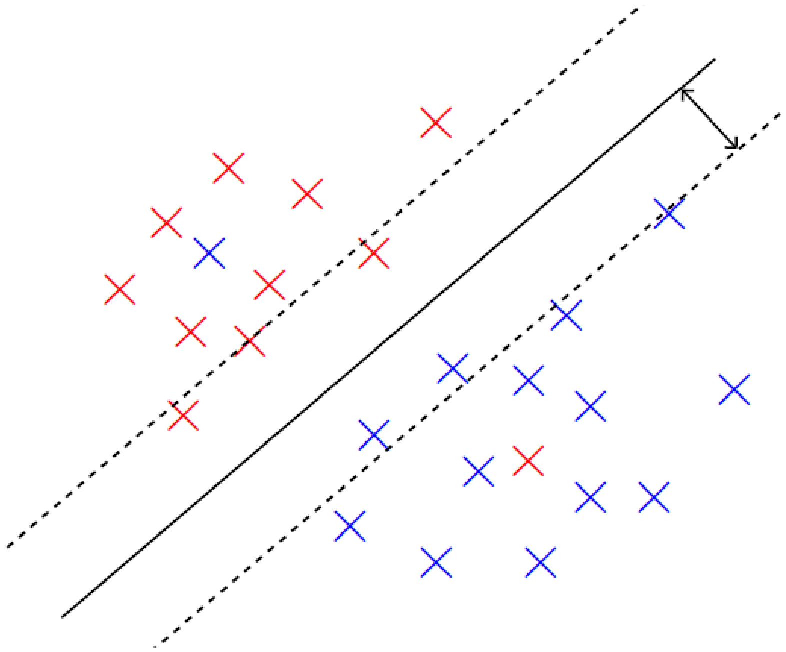
Slightly Linearly Inseparable

Allow a few points on the wrong side (slack variables)…

"Soft margin"

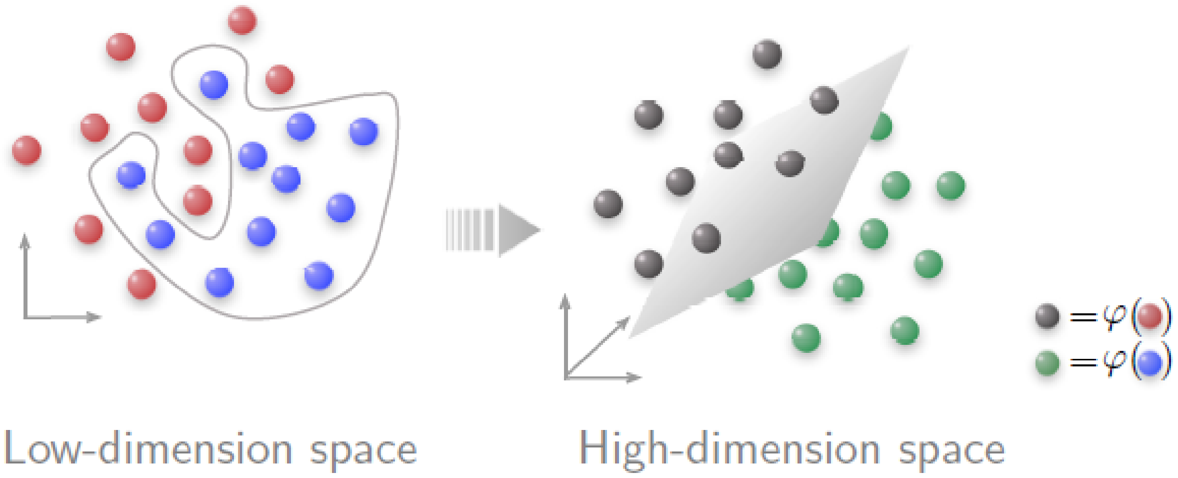
Q: Which hyperplane is the best?

A: The one that maximizes the soft "margin"!



SVM
Severely Linearly Inseparable

Map the data into a new space, then apply linear SVM

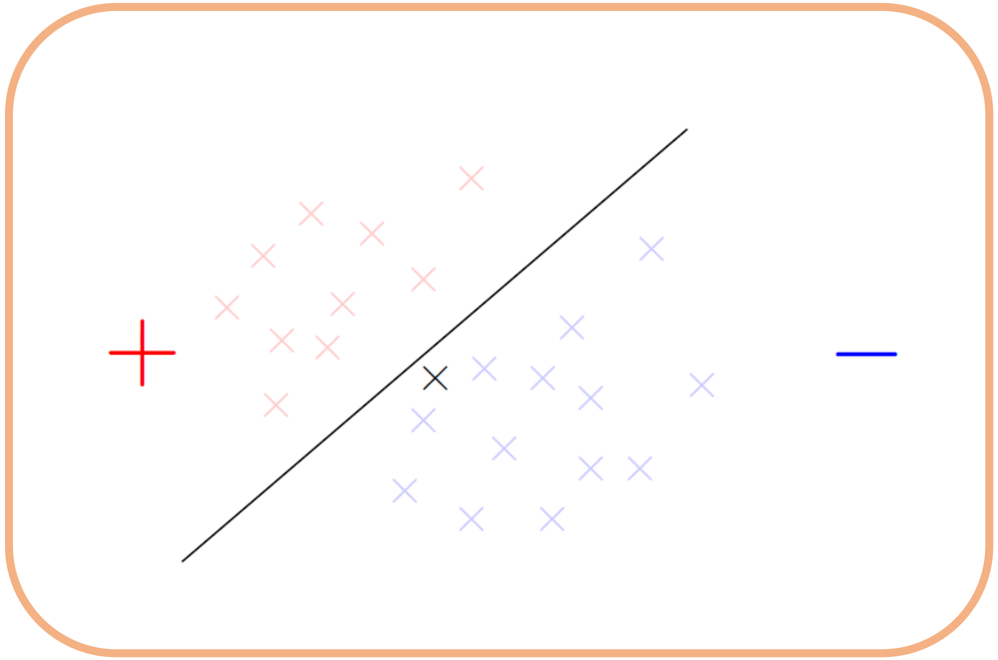


"Kernel"

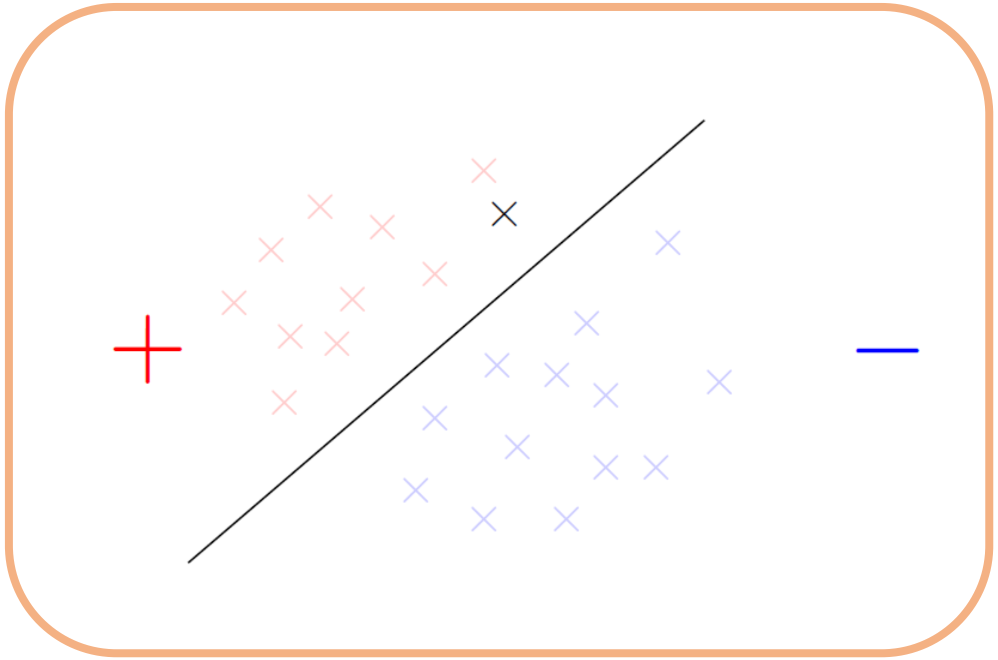
Classifier Evaluation

Let's assume that we are done with training of SVM ☺

Question: What should be the label of these points?



(a)

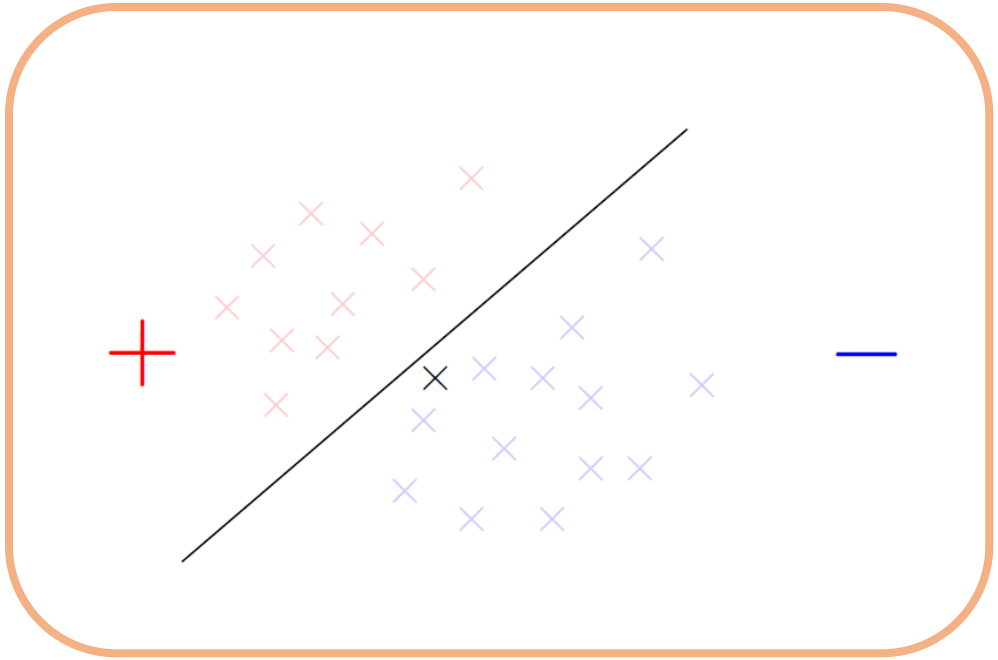


(b)

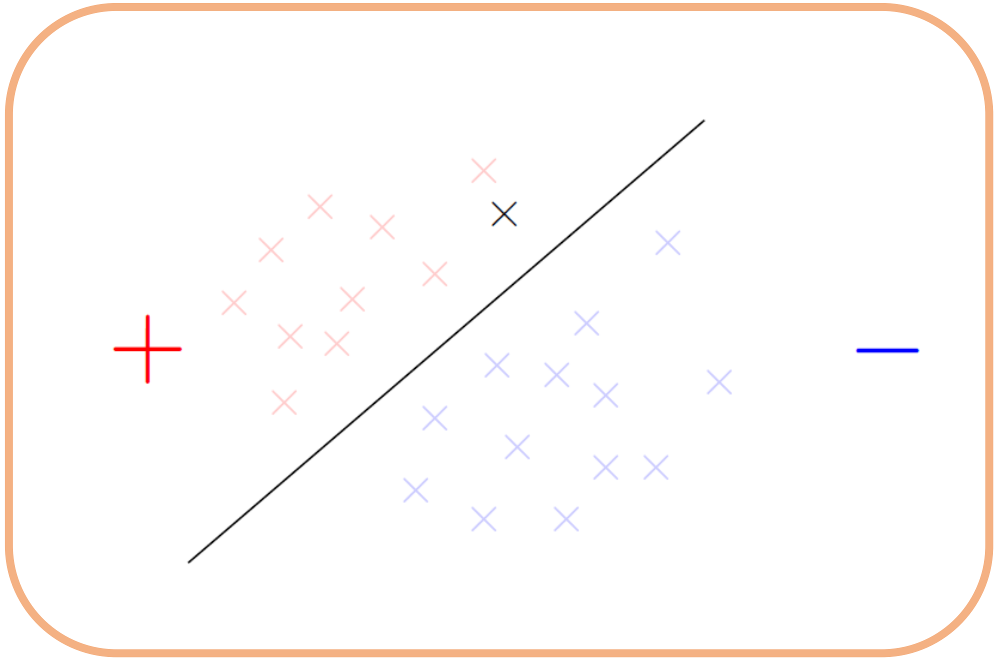
Classifier Evaluation

Let’s assume that we are done with training of SVM ☺

Question: What should be the label of these points?



(a) minus

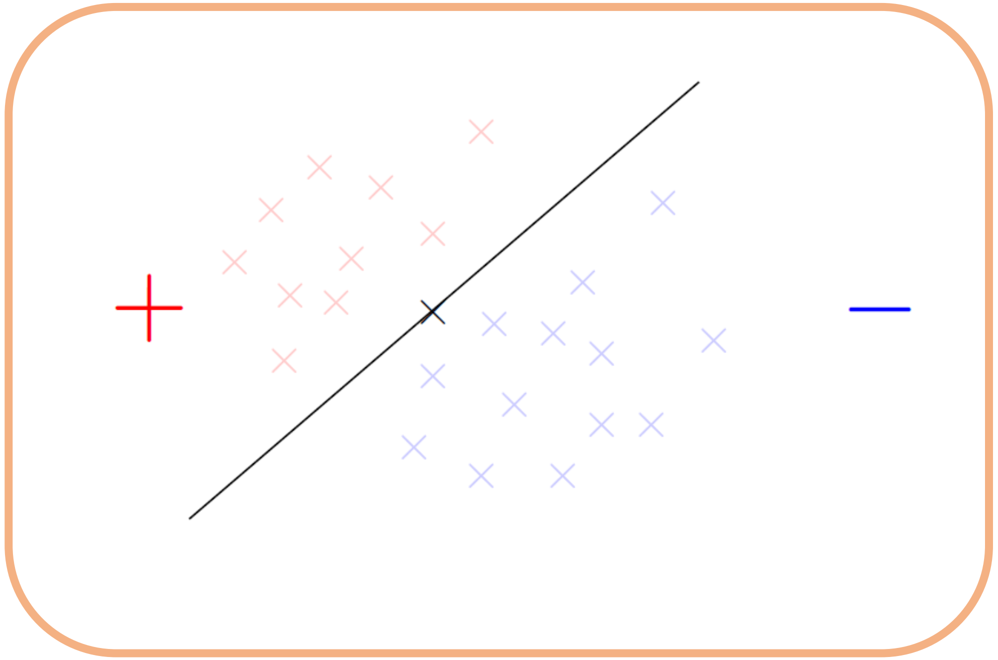


(b) plus

Linear Classification
Classifier Evaluation

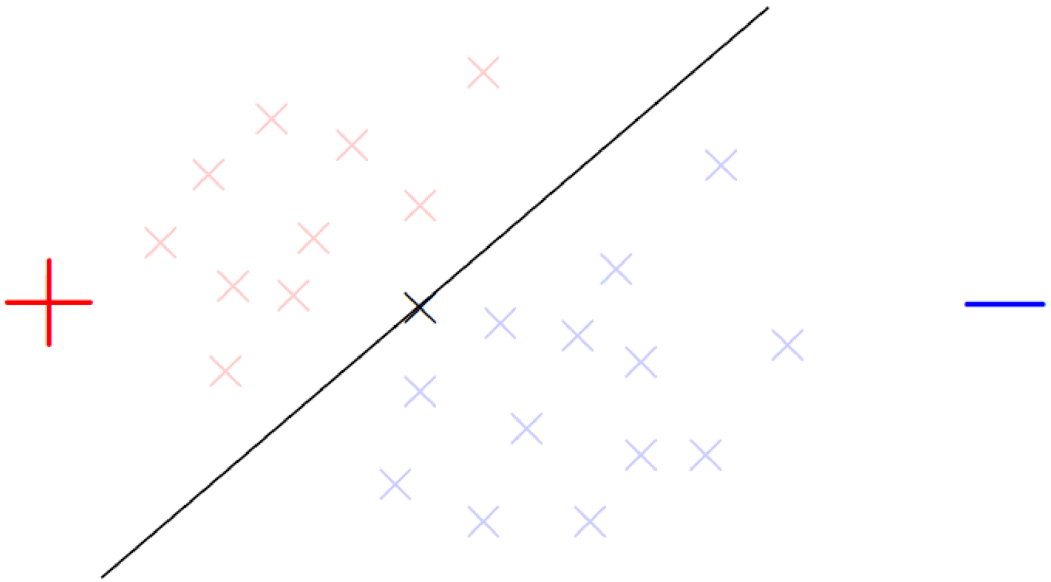
Let’s assume that we are done with training of SVM ☺

Questions: How about this point?



Is it possible to introduce the notion of confidence/probability score into the model?

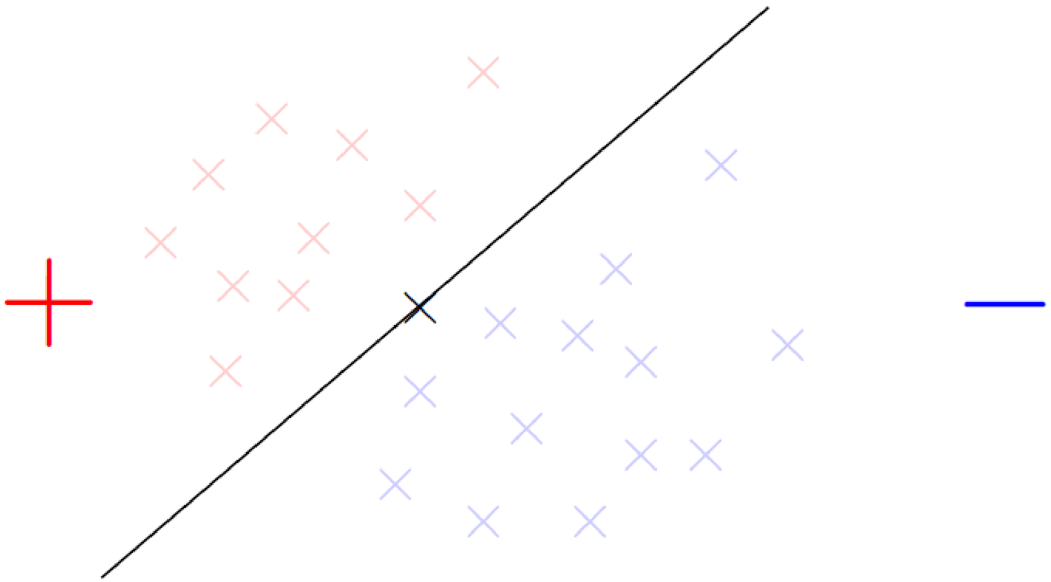
“Linear classification: classification with probability”



50% positive, 50% negative

Is it possible to introduce the notion of confidence/probability score into the model?

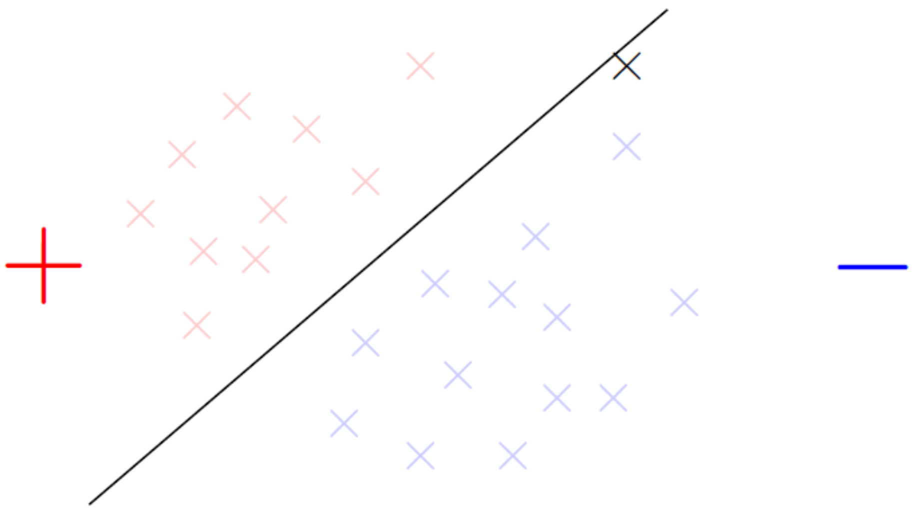
“Linear classification: classification with probability”



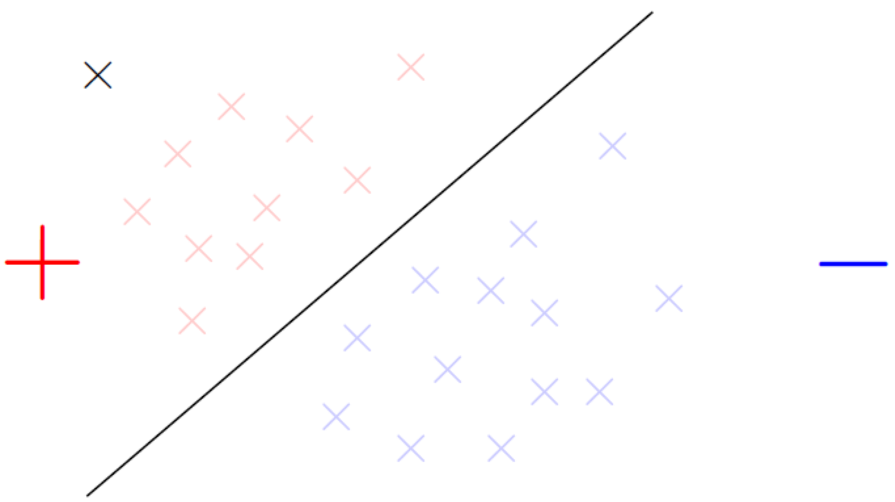
50% positive, 50% negative

Is it possible to introduce the notion of confidence/probability score into the model?

“Linear classification: classification with probability”



45% positive, 55% negative



80% positive, 20% negative

LOGISTIC REGRESSION

Is it possible to introduce the notion of confidence/probability score into the model?

Answer:

YES! Logistic Regression…

Big Picture & Motivation

- Support Vector Machines (with me)
- Linear Regression

So where is logistic regression in this picture?

From Linear Regression to Logistic Regression

Recall

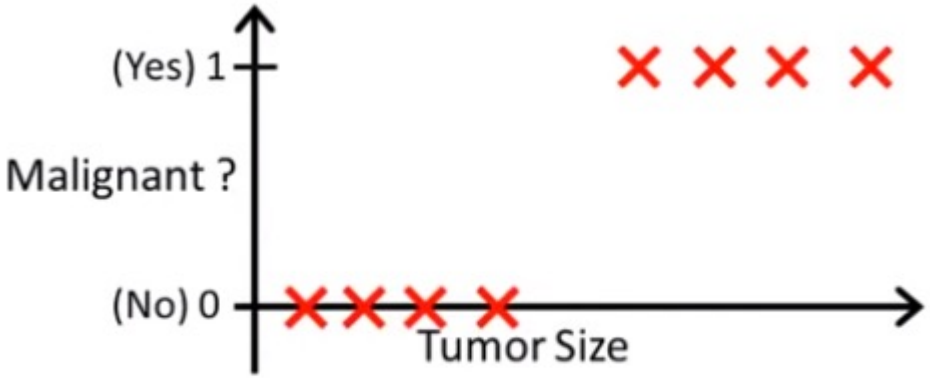
Classification

Tumor: malignant/benign

How do we develop a classification algorithm?

$y \in \{0,1\}$

0: "negative class" (benign tumor), 1: "positive class" (malignant tumor)

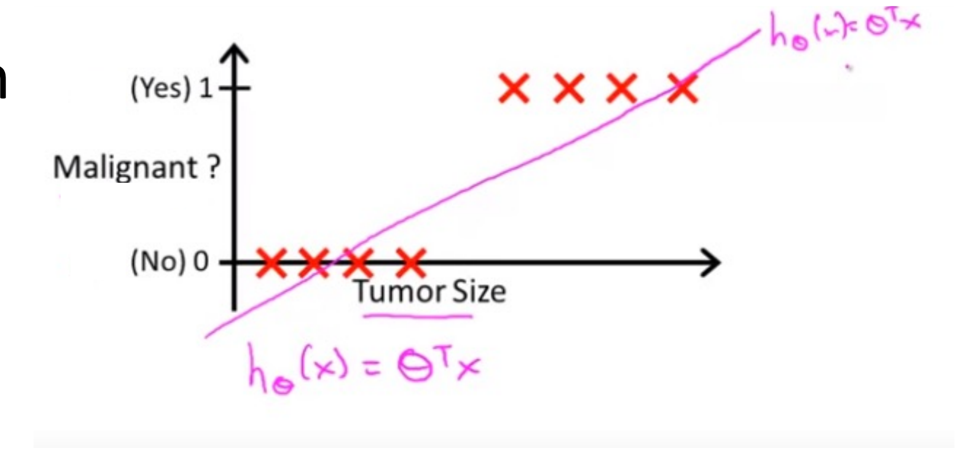


An example of a training set for classification task

This page is partially taken from Stanford lecture notes. If you do not remember linear regression, please go back and study.

Classification with Linear Regression

Given this training set, apply linear regression and try to fit the data into straight line.
The hypothesis looks like:

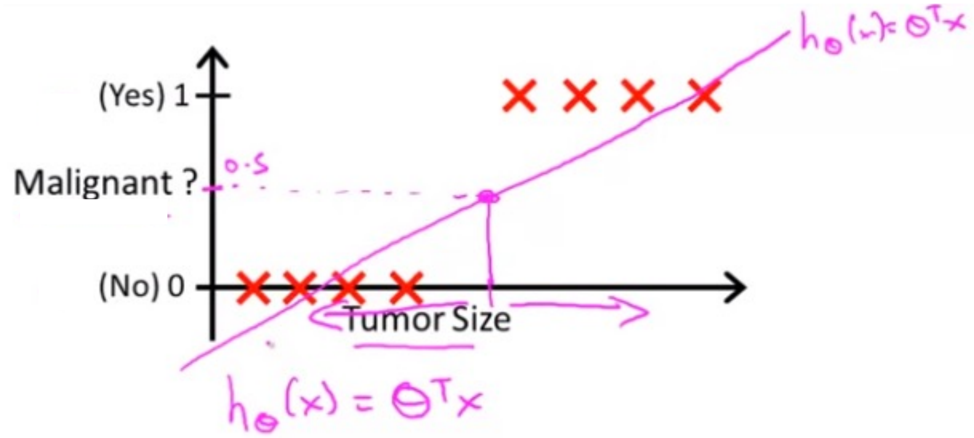


<u>To make prediction:</u>

Threshold the classifier output at 0.5

$h_\theta(x) \geq 0.5$ predict $y = 1$

$h_\theta(x) < 0.5$ predict $y = 0$

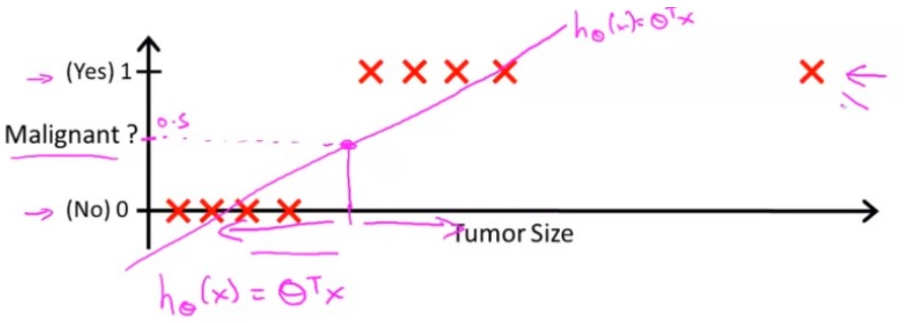


☺ It looks like linear regression can classify the data!

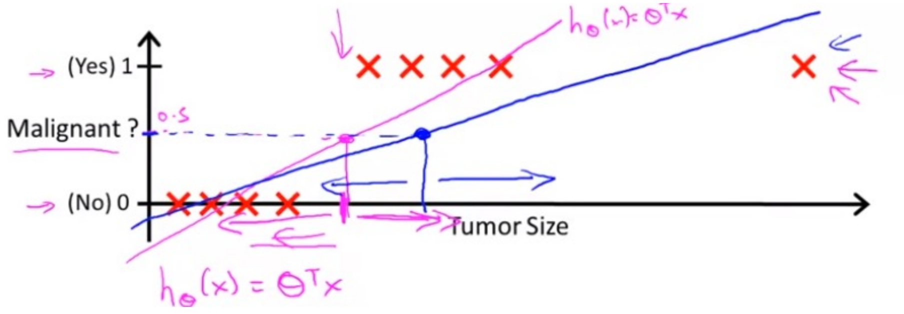
This page is partially taken from Stanford lecture notes. If you do not remember linear regression, please go back and study.

Classification with Linear Regression

Let's add one more point to the training data…



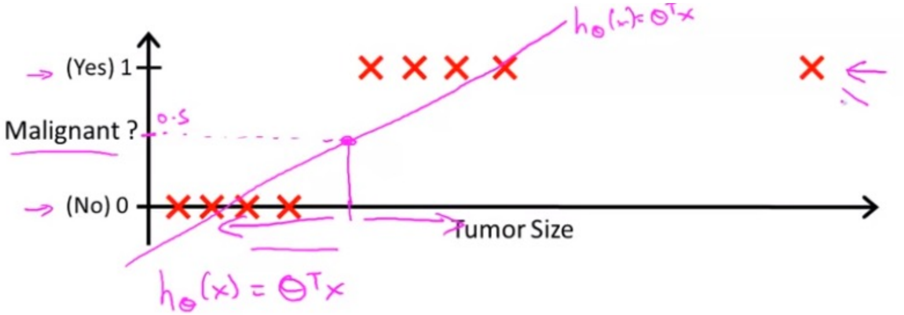
If we run linear regression with the new data:



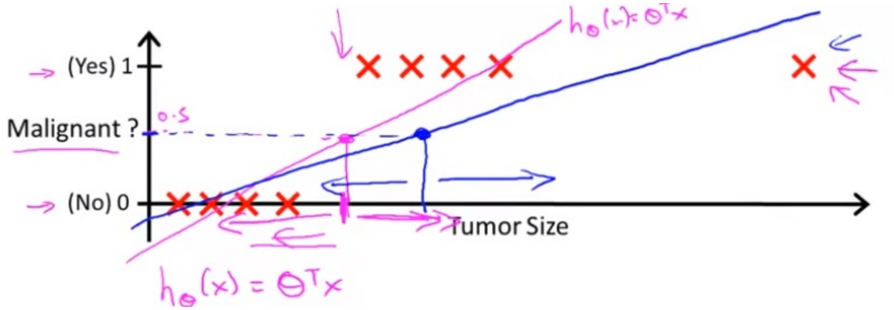
This page is partially taken from Stanford lecture notes. If you do not remember linear regression, please go back and study.

Classification with Linear Regression

Let’s add one more point to the training data…



If we run linear regression with the new data:



It is a bad classification… By adding one example, we can decrease the accuracy a lot. So, linear regression is often not a good classification method. Previously, linear regression was lucky!

This page is partially taken from Stanford lecture notes. If you do not remember linear regression, please go back and study.

Logistic Regression

Intuition & basic definition…

Logistic regression is actually a classification algorithm…

- We would like to have a classifier that outputs values between 0 and 1:

$$0 \leq h_\theta(x) \leq 1$$

(classification: $y = 0$ or $y = 1$)

Linear regression: $h_\theta(x) = \theta^T x$

Logistic regression: $h_\theta(x) = g(\theta^T x)$

where $\boxed{g(z) = \frac{1}{1+e^{-z}}}$

Sigmoid function = logistic function

Hypothesis function:

$$\boxed{h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}}$$

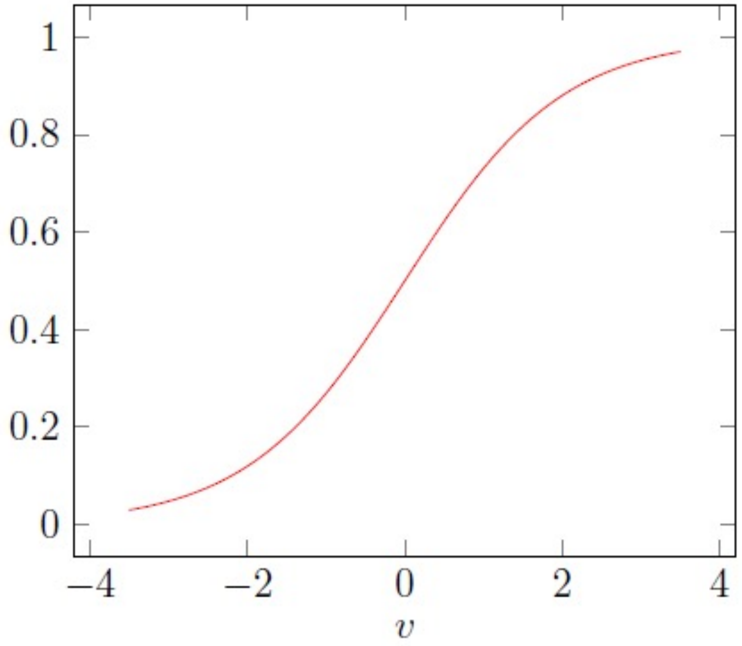
Logistic regression

<u>Why sigmoid/logistic function?</u>

1. Maps any number in between 0 and 1.
2. Interpret the results as a probability

The cost function is constructed to maximize the probability of correct classification.

Easy to work with!



In logistic regression, given training set, we will find the parameters θ.
Before discussing how to estimate these parameters, let's talk the interpretation of this model.

Logistic Regression

Interpretation

$h_\theta(x)$ = the estimated probability that $y = 1$ on input x. Let’s see an example together.

Logistic Regression

Interpretation

$h_\theta(x)$ = the estimated probability that $y = 1$ on input x. Let's see an example together.

$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ Tumor\ size \end{bmatrix}$

$h_\theta(x)$=0.7 → tell the patient that 70% probability for tumour to be malignant (sadly)

$$h_\theta(x) = p(y = 1|x;\theta) \quad \text{"}probability\ that\ y = 1, given\ x, parameterized\ by\ \theta\text{"}$$

Note that y can only be 0 or 1 → $p(y = 1|x;\theta)+p(y = 0|x;\theta) = 1$

Logistic Regression

How the hypothesis function looks like?

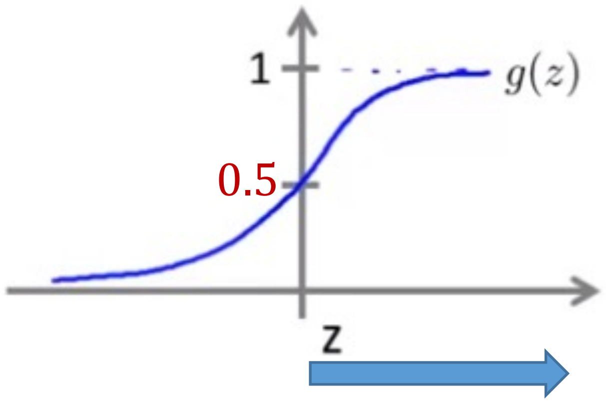
Decision Boundary…

Logistic regression – decision boundary

Hypothesis function $\Rightarrow$ $$h_\theta(x) = g(\theta^T x)$$

$$\theta^T x = z$$

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



$$h_\theta(x) = p(y = 1 | x, \theta)$$

$g(z) \geq 0.5$ **if** $z \geq 0$

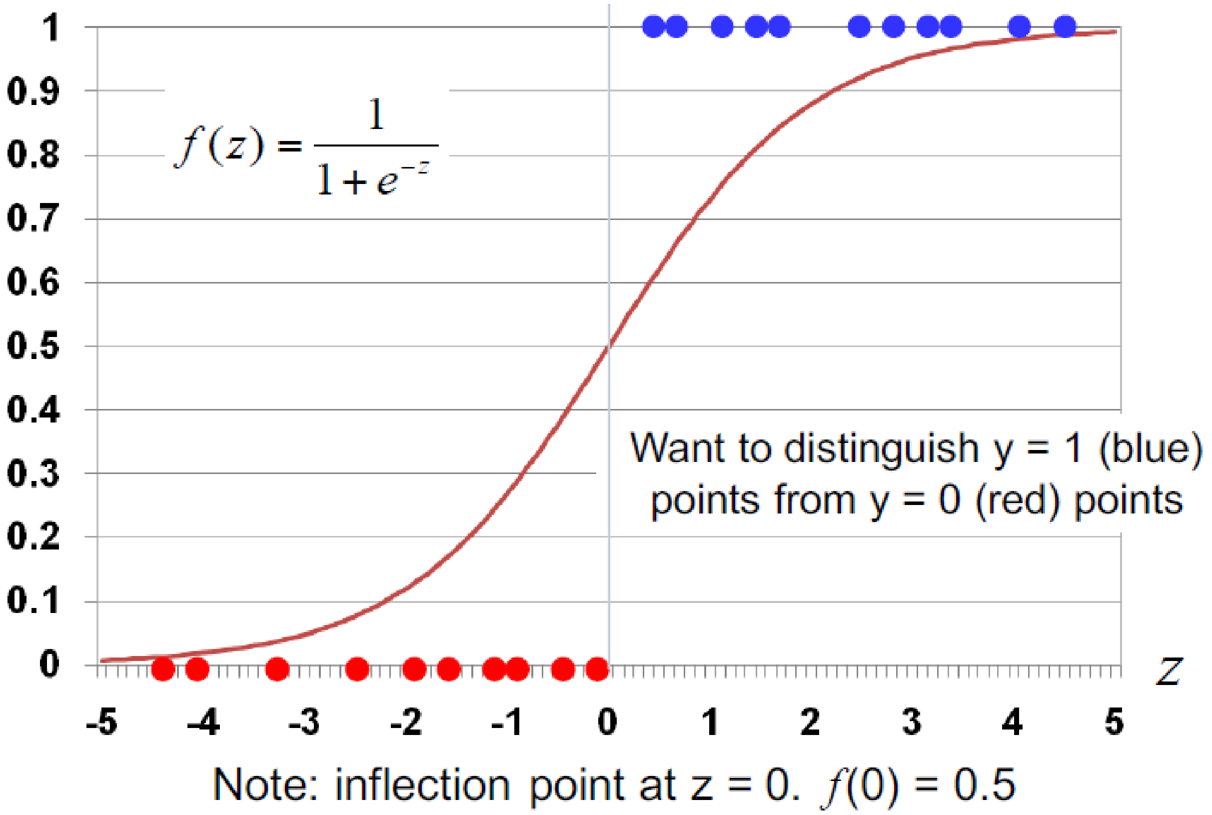
$h_\theta(x) = g(\theta^T x) \geq \mathbf{0.5}$
whenever $z = \theta^T x \geq 0$

Threshold the classifier output at 0.5

$h_\theta(x) \geq 0.5$ predict $y = 1$ $\Rightarrow$ $\theta^T x \geq 0$

$h_\theta(x) < 0.5$ predict $y = 0$ $\Rightarrow$ $\theta^T x < 0$

Decision boundary example



Let's see an example…

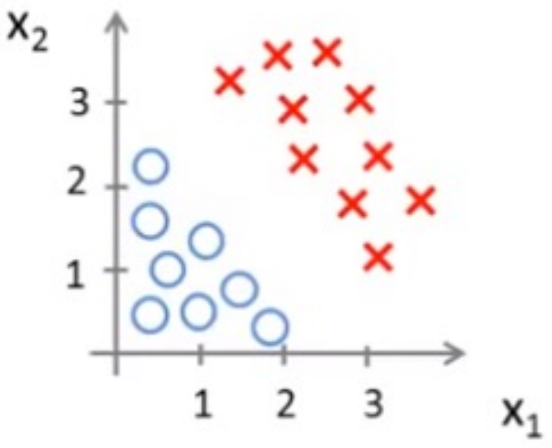
Logistic regression – decision boundary

Example 1:

Let's assume we know the parameters of the model.

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

Given that $\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$, let's try to figure out where the hypothesis ends of predicting $y = 0$ and $y = 1$.

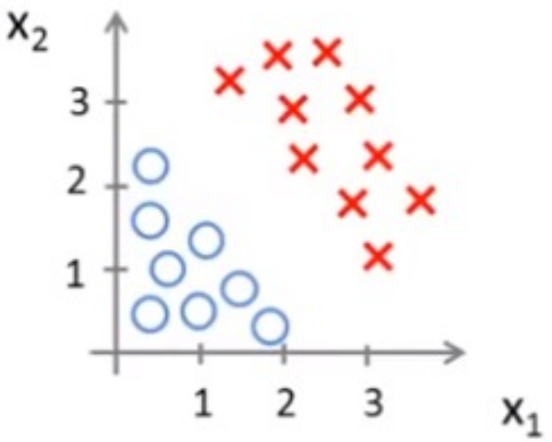


Logistic regression – decision boundary

Example 1:

Let's assume we know the parameters of the model.

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$



Given that $\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$, let's try to figure out where the hypothesis ends of predicting $y = 0$ and $y = 1$.

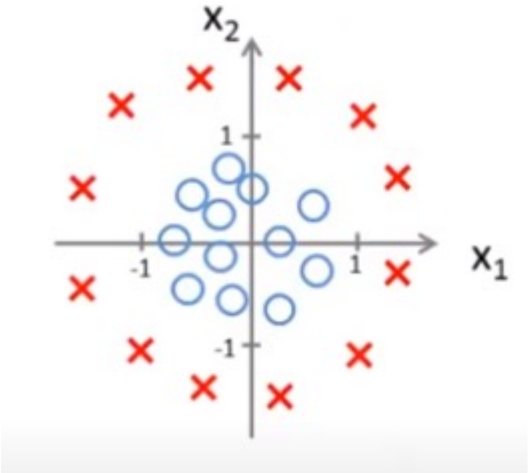
$y = 1$ if $\underbrace{\boxed{-3 + x_1 + x_2}}_{\theta^T x} \geq 0$

$y = 0$ if $\underbrace{\boxed{-3 + x_1 + x_2}}_{\theta^T x} < 0$

$y = 1$ **if** $x_1 + x_2 \geq 3$

Note that decision boundary is a property of hypothesis and the parameters…

Logistic regression – decision boundary



Example 2: Nonlinear decision boundaries

How can we fit the Logistic regression to this sort of data?

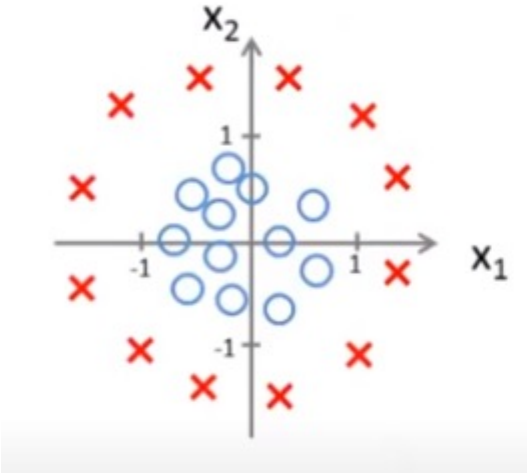
Let's assume our hypothesis is:

$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$ and $\theta = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$

Logistic regression – decision boundary

Example 2: Nonlinear decision boundaries

How can we fit the Logistic regression to this sort of data?



Let's assume our hypothesis is:

$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$ and $\theta = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$

$y = 1$ if $-1 + x_1^2 + x_2^2 \geq 0$ (equation for a circle, radius 1 & centered around origin)

$y = 0$ if $-1 + x_1^2 + x_2^2 < 0$

How does the decision boundary look like?

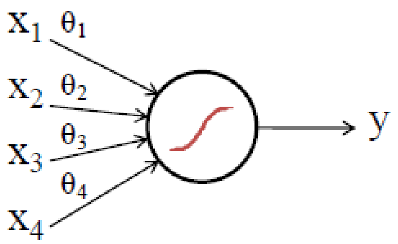
Through polynomial terms, we can actually obtain more complex decision boundaries (elliptic shapes, or some other funny shapes that can separate the data)

Logistic Regression

Overall Problem…

How to choose parameters θ?

Logistic Regression – overall problem



Training set: $(x^1, y^1), (x^2, y^2), (x^3, y^3), \ldots, (x^m, y^m)$

Training set of m examples: We'll use superscripts to refer to individual instances in the training set

$$\boxed{y \in \{0,1\},} \qquad x \in \begin{bmatrix} x_0 \\ x_1 \\ \ldots \\ x_n \end{bmatrix}, \qquad x_0 = 1$$

Each example is n+1 dimensional

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} = \frac{e^{\theta^T x + \theta_0}}{1 + e^{\theta^T x + \theta_0}}$$

h_θ is the probability of predicting the label positive (y=+1). Parameters: θ

What shall we optimize?

How to choose θ?

Recall

- In linear regression, our cost function looks like this:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}(h_\theta\left(x^i\right) - y^i)^2$$

This cost function works well for linear regression.

Recall

- In linear regression, our cost function looks like this:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}(h_\theta(x) - y)^2$$

This cost function works well for linear regression.

If we use the same cost function for logistic regression, this will be a non-convex function of the parameters θ:

- In logistic regression, $h_\theta(x)$ has nonlinearity. $\Rightarrow$ $\boldsymbol{h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}}$
- If you take this sigmoid function and plug it into $J(\theta)$ of linear regression, and plot $J(\theta)$:

Recall

- In linear regression, our cost function looks like this:

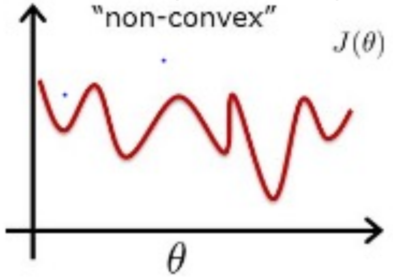
$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}(h_\theta(x) - y)^2$$

This cost function works well for linear regression.

If we use the same cost function for logistic regression, this will be a non-convex function of the parameters θ:

  - In logistic regression, $h_\theta(x)$ has nonlinearity. $\boldsymbol{h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}}$
  - If you take this sigmoid function and plug it into $J(\theta)$ of linear regression, and plot $J(\theta)$:

With many local optima!



If you run gradient descent on non-convex function, it is not guaranteed to converge to global minimum.

Recall

- In linear regression, our cost function looks like this:

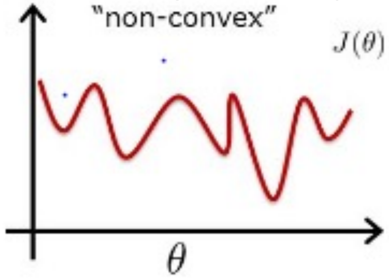
$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}(h_\theta(x) - y)^2$$

This cost function works well for linear regression.

If we use the same cost function for logistic regression, this will be a non-convex function of the parameters θ:

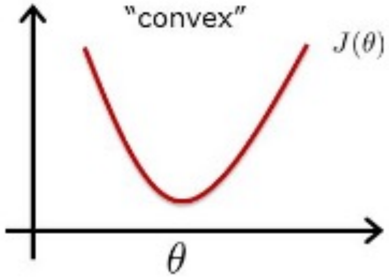
- In logistic regression, $h_\theta(x)$ has nonlinearity. $\boldsymbol{h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}}$
- If you take this sigmoid function and plug it into $J(\theta)$ of linear regression, and plot $J(\theta)$:

With many local optima!



If you run gradient descent on non-convex function, it is not guaranteed to converge to global minimum.

We hope to have:

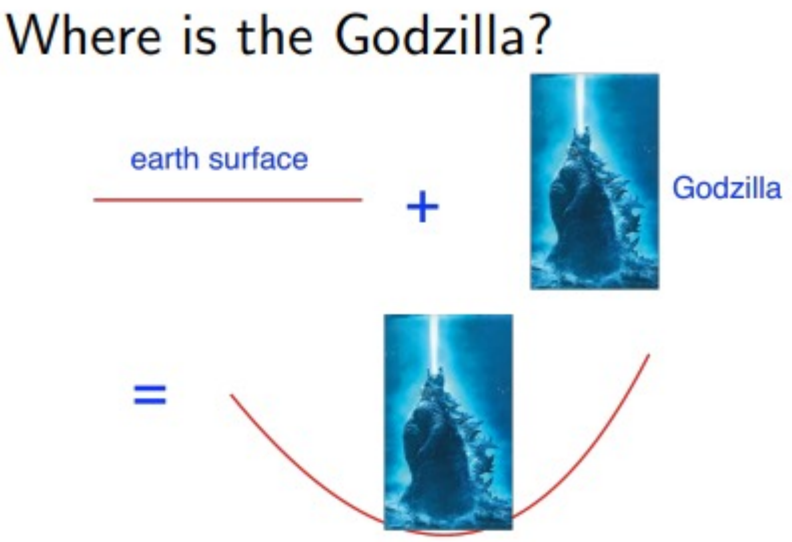


If you run gradient descent on a convex function, it will converge to global minimum.

If you want to learn more about convex optimization:

https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf (theory)

http://www.stat.cmu.edu/~ryantibs/convexopt/lectures/nonconvex.pdf (funny examples)



"convex"

Each Godzilla defines a local minima and the "heaviest" Godzilla: The global minima

"non-convex"

Restricted

Can we come up with a convex cost function for logistic regression?

Yes! ☺ Then, we can use gradient descent algorithm for optimization.

Logistic Regression: cost function

Let's study a convex cost function for logistic regression:

- $h_\theta(x)$ is a number (for example 0.7, probability of data belonging to class +1)
- Actual class label is y, and note that y is always 0 or 1. *(during training we know it!)*

Logistic regression cost function:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}) \quad \mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

It may look like a complicated function. So let's explain each part....

Logistic Regression: cost function

<u>If $y = 1$</u>

- $h_\theta(x) = 1$, then cost is equal to 0. 
- As $h_\theta(x) \to 0$, cost $\to \infty$ (we don't want this!)

<u>If $y = 0$</u>

- $h_\theta(x) = 0$, then cost is equal to 0. 
- As $h_\theta(x) \to 1$, cost $\to \infty$ (we don't want this!)

Can we simplify this cost function?

Logistic regression cost function

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases} \qquad J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

Logistic Regression: simplified cost function

Logistic regression cost function

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases} \qquad J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

Can we come up with a simpler way to write this cost function?

Rather than 2 lines, we can compress them in 1 equation. Then, we will apply Gradient Descent.

Logistic Regression: simplified cost function

Logistic regression cost function

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases} \qquad J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

Can we come up with a simpler way to write this cost function?

Rather than 2 lines, we can compress them in 1 equation. Then, we will apply Gradient Descent.

Logistic regression cost function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$= -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right)\right]$$

Logistic Regression: simplified cost function

Logistic regression cost function

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases} \qquad J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

Can we come up with a simpler way to write this cost function?

Rather than 2 lines, we can compress them in 1 equation. Then, we will apply Gradient Descent.

Logistic regression cost function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$
$$= -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right)\right]$$

Why this particular cost function?

- **It can be derived from statistics, using maximum likelihood estimation…**
- **It is convex. ☺**
- **It is the cost function everyone uses for logistic regression models.**

Logistic Regression: simplified cost function

Training:

We'll try to find parameters θ that minimizes $J(\theta)$. → Get θ

Logistic regression cost function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$
$$= -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

Run-time:

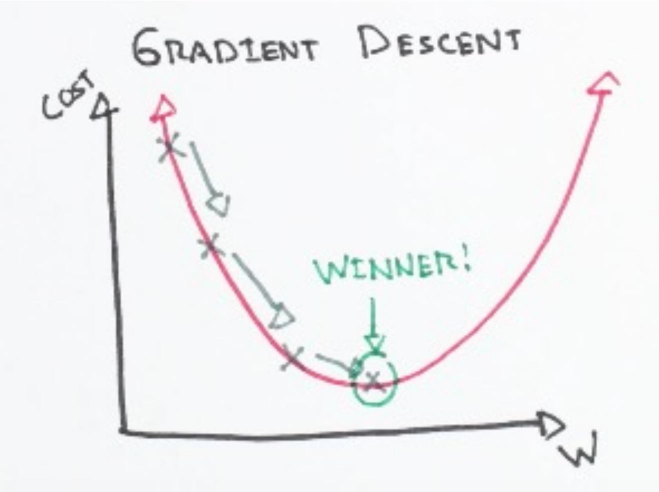
To make a prediction on a given new x assuming that we already obtained θ.

The output of the hypothesis will be interpreted as $p(y = 1|x; \theta)$

How to actually minimize $J(\theta)$?

Gradient Descent

- Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient.
- In machine learning, we use gradient descent to update the parameters of our model.



Logistic Regression: Gradient Descent

We'll use gradient descent to minimize our cost function.
Here is the usual template of gradient descent.
We will update the parameters by taking the derivative of the function.

Gradient Descent

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log (1 - h_\theta(x^{(i)}))\right]$$

Want $\min_\theta J(\theta)$:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

(simultaneously update all θ_j)

$\}$

How to take this derivative?
You can try at home.
If you don't know the answer, don't worry about it.

Logistic Regression: gradient descent

- If we take the derivative and plug into the equation:

Gradient Descent

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right)\right]$$

Want $\min_\theta J(\theta)$:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update all θ_j)

$\}$

The definition of hypothesis has changed!

Linear regression: $h_\theta(x) = \theta^T x$

Logistic regression: $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

A surprising fact! Looks identical to linear regression gradient descent update rule!

Restricted

Logistic Regression – Another perspective

Logistic Regression – Another perspective

So far, we always assumed that our labels are 0 and 1.

What happens if our labels are not 0 and 1, but instead -1 and +1?

Class labels are +1 and -1

A new cost function!

Another way of writing the cost function…

Logistic regression – Objective Function?

h_θ **is the probability of predicting the label positive (y=+1)**

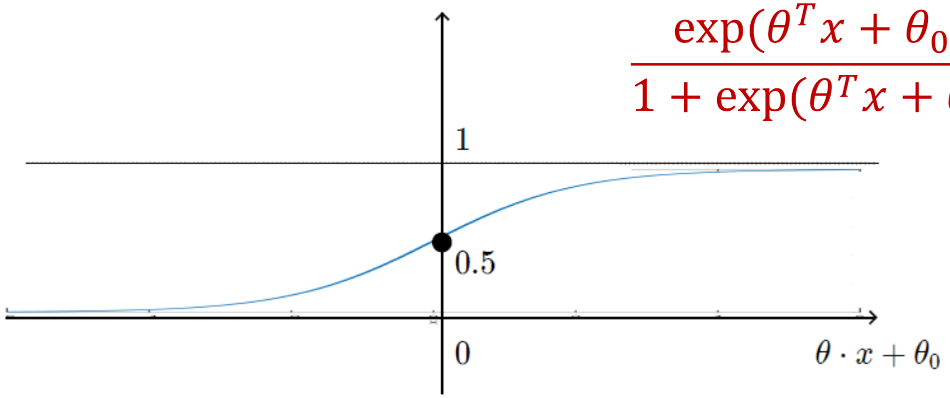
$$p(y|x) = \begin{cases} h_\theta(x) & for\ y = +1 \\ 1 - h_\theta(x) & for\ y = -1 \end{cases}$$

Logistic regression – Objective Function?

h_θ **is the probability of predicting the label positive (y=+1)**

$$p(y|x) = \begin{cases} h_\theta(x) & for\ y = +1 \\ 1 - h_\theta(x) & for\ y = -1 \end{cases}$$

$$\frac{\exp(\theta^T x + \theta_0)}{1 + \exp(\theta^T x + \theta_0)}$$



$$p(y = +1|x) = \frac{\exp(\theta^T x + \theta_0)}{1 + \exp(\theta^T x + \theta_0)} = \delta(\theta^T x + \theta_0)$$

$$p(y = -1|x) = \frac{1}{1 + \exp(\theta^T x + \theta_0)} = \frac{\exp(-(\theta^T x + \theta_0))}{1 + \exp(-(\theta^T x + \theta_0))}$$

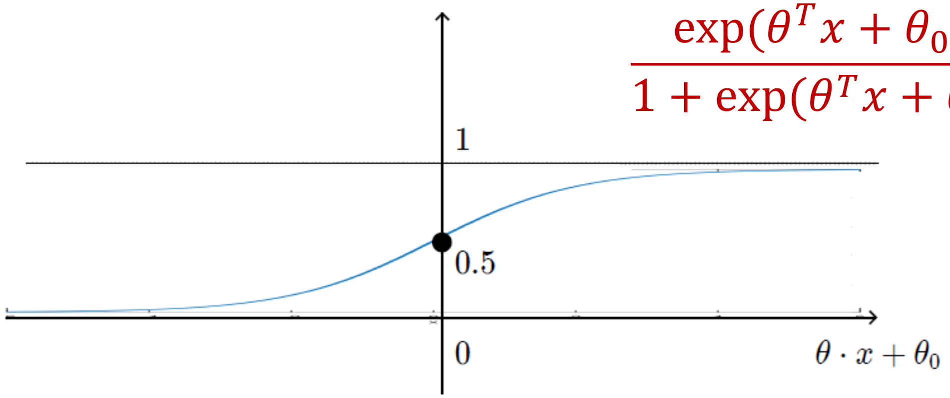
$$= \delta(-(\theta^T x + \theta_0))$$

Logistic regression – Objective Function?

h_θ **is the probability of predicting the label positive (y=+1)**

$$p(y|x) = \begin{cases} h_\theta(x) & for\ y = +1 \\ 1 - h_\theta(x) & for\ y = -1 \end{cases}$$

$$\frac{\exp(\theta^T x + \theta_0)}{1 + \exp(\theta^T x + \theta_0)}$$



$$p(y = +1|x) = \frac{\exp(\theta^T x + \theta_0)}{1 + \exp(\theta^T x + \theta_0)} = \delta(\theta^T x + \theta_0)$$

$$p(y = -1|x) = \frac{1}{1 + \exp(\theta^T x + \theta_0)} = \frac{\exp(-(\theta^T x + \theta_0))}{1 + \exp(-(\theta^T x + \theta_0))}$$

$$= \delta(-(\theta^T x + \theta_0))$$

$$p(y|x) = \delta(y(\theta^T x + \theta_0))$$

For classification we care about $p(y \mid x)$

Logistic regression – Objective Function?

- Could we model $P(y \mid x)$ directly? Welcome our friend, logistic regression!

Training set examples: $(x^1, y^1), (x^2, y^2), (x^3, y^3), \ldots, (x^n, y^n)$

$$\max_{\theta,\theta_0} \quad \prod_{i=1}^{n} p(y^{(i)}|x^{(i)})$$

$$\max_{\theta,\theta_0} \quad \log \prod_{i=1}^{n} p(y^{(i)}|x^{(i)})$$

$$\max_{\theta,\theta_0} \quad \sum_{i=1}^{n} \log p(y^{(i)}|x^{(i)})$$

$$\min_{\theta,\theta_0} \quad \sum_{i=1}^{n} \log 1/p(y^{(i)}|x^{(i)})$$

Logistic regression – Objective Function

What shall we optimize?

Training set: $(x^1, y^1), (x^2, y^2), (x^3, y^3),\ldots., (x^n, y^n)$

$$\sum_{i=1}^{n} \log 1/p(y^i|x^i)$$

Loss Function:

$$\sum_{i=1}^{n} \log(1 + \exp\left(-y^i(\theta^T x^i + \theta_0)\right))$$

Logistic regression – Objective Function

What shall we optimize?

Training set: $(x^1, y^1), (x^2, y^2), (x^3, y^3), \ldots, (x^n, y^n)$

$$\sum_{i=1}^{n} \log 1/p(y^i|x^i)$$

Loss Function:

$$\sum_{i=1}^{n} \log(1 + \exp\left(-y^i(\theta^T x^i + \theta_0)\right))$$

(1) Homework question: What is the benefit of using logarithm? Why is this expression computationally more "convenient"?
(2) We can iteratively optimize this problem using stochastic gradient descent.

Logistic regression – Learning

Note that while doing stochastic gradient descent, we need to consider the objective function associated with each instance.

The objective associated with the t^{th} instance is:

Let us drop θ_0 for now:

$$e^{(t)}(\theta) = \log\left(1 + \exp(-y^{(t)}(\theta \cdot x^{(t)})\right)$$

$$\nabla e^{(t)}(\theta) = \frac{-y^{(t)}x^{(t)}}{1+\exp(y^{(t)}(\theta \cdot x^{(t)}))}$$

$$\theta \leftarrow \theta - \eta \nabla e^{(t)}(\theta)$$

η is the magnitude of the step size that we take. If you keep updating θ using the equation above, you will converge on the best values of θ. You now have an intelligent model.

Logistic regression – Learning

Note that w … bjective
function ass…

The objectiv…

The learning rate η is a hyperparameter that must be adjusted. If it's too high, the learner will take steps that are too large, overshooting the minimum of the loss function. If it's too low, the learner will take steps that are too small, and take too long to get to the minimum.

If you're interested to learn more -> DL course

$$\theta \leftarrow \theta - \eta \nabla e^{(t)}(\theta)$$

η is the magnitude of the step size that we take. If you keep updating θ using the equation above, you will converge on the best values of θ. You now have an intelligent model.

Logistic Regression

Assume now we have already learned our model parameters in the training phase. We now would like to make predictions for the new input x. What shall we do?

Logistic Regression - How shall we predict the output label?

- Now we have a new input x

$$\frac{p(y=+1|x)}{p(y=-1|x)} > 1 \ ?$$

If yes, positive, otherwise negative!

$$\log \frac{p(y=+1|x)}{p(y=-1|x)} > 0 \ ?$$

If yes, positive, otherwise negative!

$$p(y=+1|x) \vee \ ? \ p(y=-1|x)$$

If yes, positive, otherwise negative!

Logistic Regression - How shall we predict the output label?

- Let us take a closer look at this:

$$\frac{\exp(\theta^T x + \theta_0)}{1 + \exp(\theta^T x + \theta_0)}$$

$$\log \frac{\boxed{P(y = +1|x)}}{\boxed{P(y = -1|x)}} = \log \exp(\theta \cdot x + \theta_0) = \theta \cdot x + \theta_0$$

$$\frac{1}{1 + exp(\theta^T x + \theta_0)}$$

Logistic Regression - How shall we predict the output label?

- Let us take a closer look at this:

$$\frac{\exp(\theta^T x + \theta_0)}{1 + \exp(\theta^T x + \theta_0)}$$

$$\log \frac{\boxed{P(y=+1|x)}}{\boxed{P(y=-1|x)}} = \log \exp(\theta \cdot x + \theta_0) = \theta \cdot x + \theta_0$$

$$\frac{1}{1 + exp(\theta^T x + \theta_0)}$$

Now, we can see that this is a linear function. What does this mean?
It shows that the decision boundary for the logistic regression is a linear function…

Logistic Regression

or SVM?

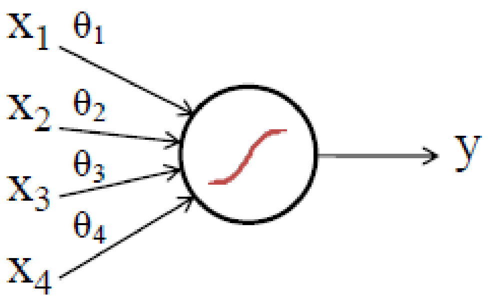
Logistic Regression vs SVM

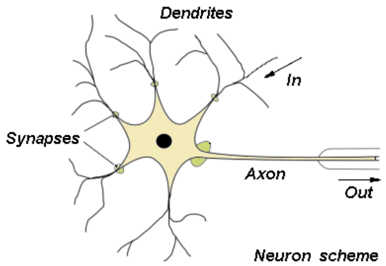
- **Logistic regression** focuses on **maximizing the probability of the data**. The farther the data lies from the separating hyperplane (on the correct side), the happier LR is.

- SVM tries to find the separating hyperplane that maximizes the distance of the closest points to the margin (the support vectors). If a point is not a support vector, it doesn't really matter.

Logistic Regression vs Neural Networks

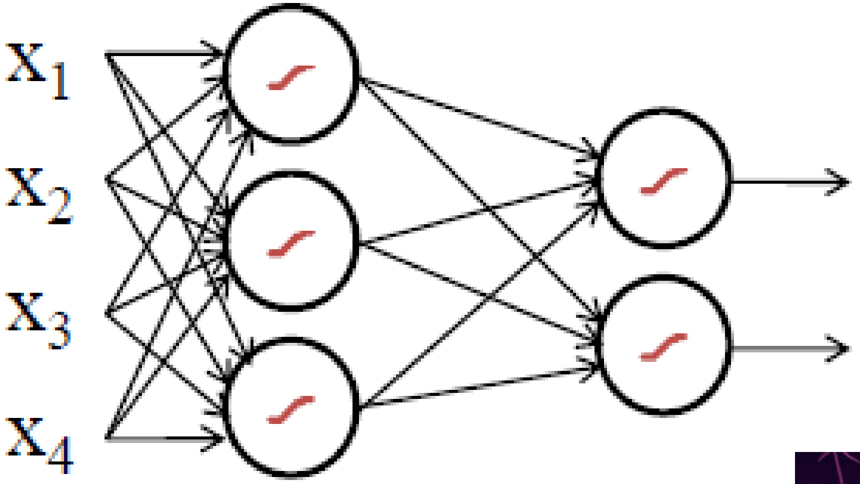
Logistic Regression:



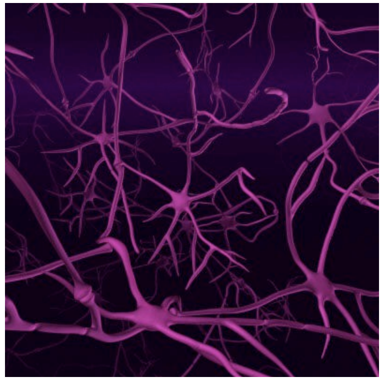


A neuron

Neural Network:



Brain



Logistic regression is same as a one node neural network! A neural network can be viewed as a series of logistic regression classifiers stacked on top of each other …

Awesome classifier, terrible name ☺

Regression Algorithms	Classification Algorithms
Linear Regression	Naïve Bayes
	Logistic Regression

Logistic regression is the building block of artificial neural networks!
Logistic classifier? ☺

Restricted

Conclusion

- What is logistic regression? When should we use it?
- The intuition of logistic regression? what type of problems logistic regression can solve?
- Logistic regression vs SVM
- The decision boundary of logistic regression
- How to perform learning and prediction under logistic regression?

MUST: Please study the lecture notes on logistic regression

Suggestion: If you're not very clear or want to learn more, please do read: Logistic regression part of "C. Bishop: Pattern Recognition and Machine Learning. Springer, 2006" *(recommended text book).*

THE END