The example to combine evolutionary algorithm and machine learning

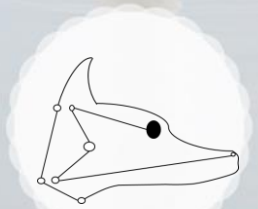
Using logit model and particle swarm optimization

Markliou
10/23/2017

About the speaker

- Education
  - Bioinformatics and Systems Biology
  - Biotechnology
  - Life Science

- Experiences
  - Data Scientist, Light Up Biotech. Corp.
  - Machine learning consultant, Bcondux Corp.
  - Algorithm Engineer, 京悅投資開發股份有限公司
  - Postdoc fellow, NCTU
  - Research Assistant, NCTU
  - 桃園市106年資訊組長出階及進階研習計畫 (Docker 助教)

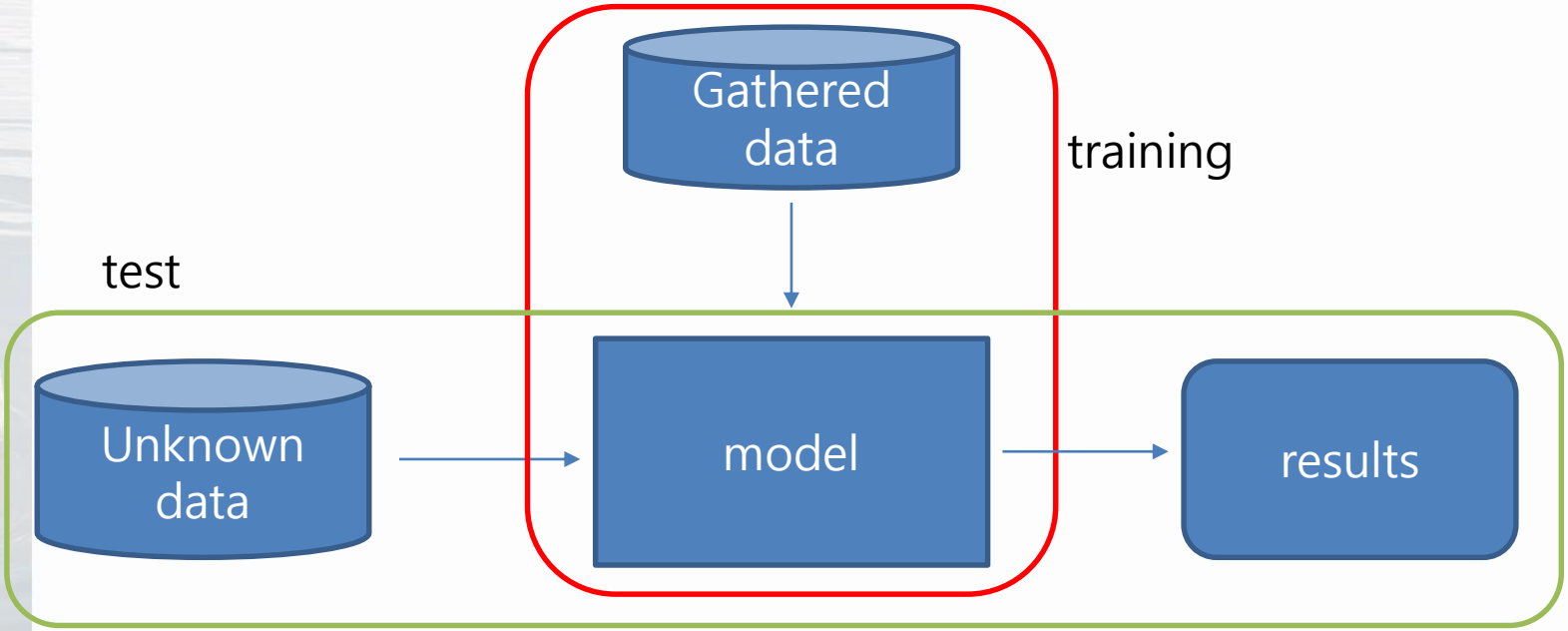


Outline

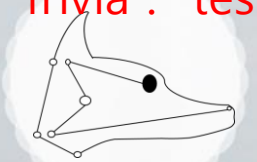
- The concept of machine learning
  - Linear regression -- Logistic regression
- The concept of evolutionary algorithm
  - Particle swarm optimization
- Combine the machine learning with evolutionary algorithm
  - Using the logit model and PSO as example
  - Aims to use PSO to solve the parameters of logit model

The concept of machine learning

1. Users have data
2. Users want to predict the unknown data



Trivia : "test" is not "testing"



Using existing data to find the relation between variables?

It's statistics

Why logistic regression?

Some concepts in machine learning is also from the statistics.

They are quite similar!!

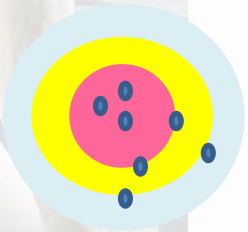
You can find the logit models in:
1. Traditional statistics – ex. Survival analysis
2. Machine learning

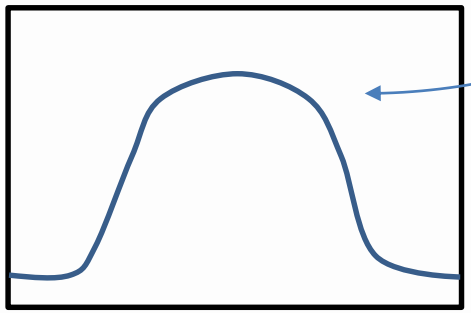
Logistic regression – from statistic

Set: All the dependent variables are linear
continuous : least square errors
categorical : continuous **with some errors**

$Y = a1x1 + a2x2 + ... + anxn + b$

Ideally



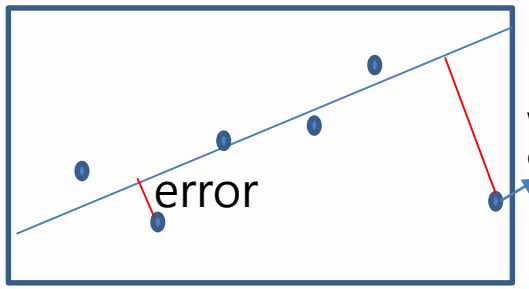


Normal distribution

$Y = a1x1 + a2x2 + ... + anxn + b$

Never forget : the least square assume the **normal distribution**

This concept make a straight line



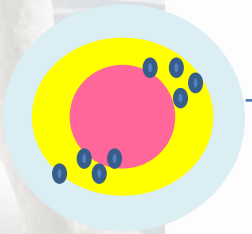
Extreme error would not often happen

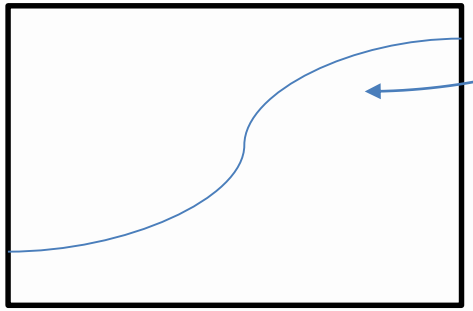
Logistic regression – from statistic

Set: All the dependent variables are linear
continuous : least square errors
categorical : continuous **with some errors**

$Y = a1x1 + a2x2 + ... + anxn + b$

Classification case





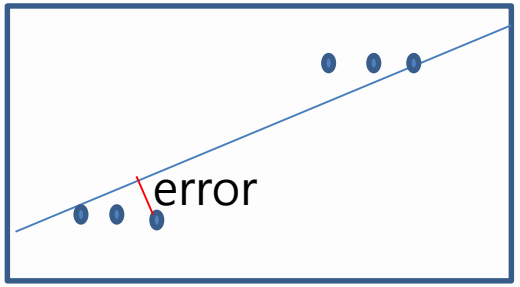
NOT
Normal distribution

$Y = a1x1 + a2x2 + ... + anxn + b$

If you want to fit this line ....

This concept also make a straight line
It will be like this ....

It is obviously : the errors between the line and data
won't be normal distribution

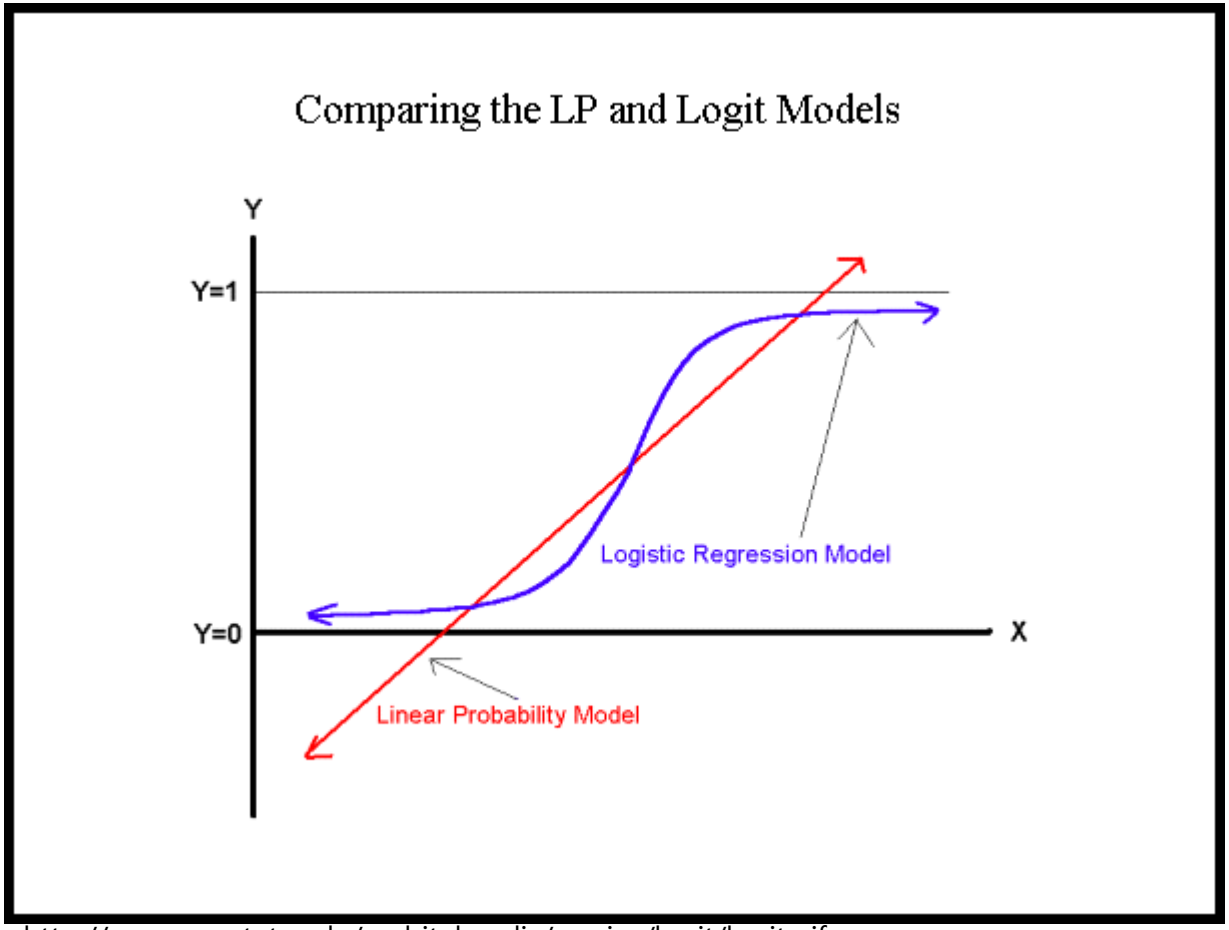


That's why the classification problem never use MSE as loss
(you can use MSE, but it will make a tragedy)

Logistic regression – from statistic

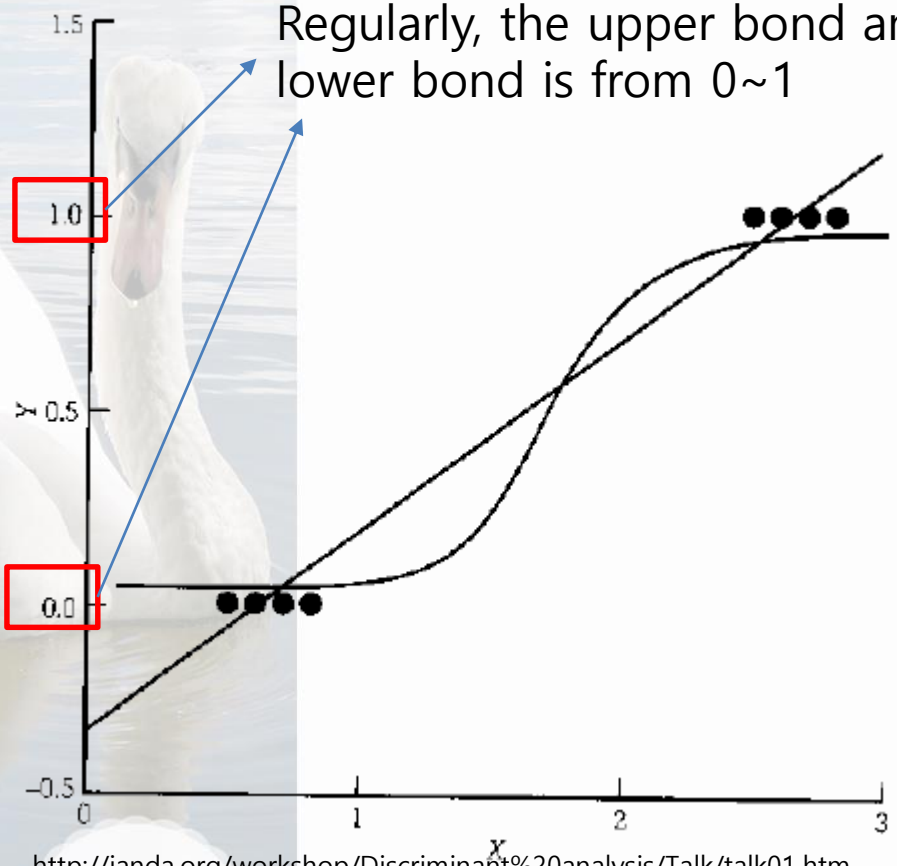
To solve this problem, we already know the distribution of data will not be normal distribution…

SO……



http://www.appstate.edu/~whiteheadjc/service/logit/logit.gif

Logistic regression – from statistic



Regularly, the upper bond and lower bond is from 0~1

Why not use "ratio"?

$$log(P/(1-P)) = \sum_{k=i}^{n} a_i x_i + b$$

case

Control
(total samples except the cases)

Odds ratio

http://janda.org/workshop/Discriminant%20analysis/Talk/talk01.htm

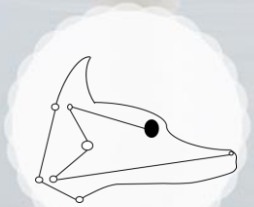
According to "maximum likelihood", using the optimization method can get the odds ratio

1. Newton
2. Gradient decent

Maximum likelihood V.S. cross entropy

- In machine learning, the loss usually use cross entropy
- In statistics, the loss usually use maximum likelihood
- But Don't worry, they are similar …

Alarm !!!!! Math time~~



The relation between BCE and ML

- BCE = binary cross entropy ML = maximum likelihood
- Set : the problem is simple as **bi-classification**
  - The ML can applied as Bernoulli

Bernoulli

$$p(y|\theta) = \prod_{i=1}^{n} \theta_i^{yi} (1 - \theta_i)^{1-yi}$$

This is a distribution from model, so..
Let p(x|θ′) denote the training model θ

$$p(y|x, \theta') = \prod_{i=1}^{n} p\theta'(y|x_i)^{yi} (1 - p\theta'(y|x_i))^{1-yi}$$

How about give a 'log' ?

$$f(\theta; x, y) = \sum_{i=1}^{n} y_i \, log \, p_{\theta'}(y|xi) + (1 - y_i) \log(1 - p_{\theta'}(y|xi))$$

$$\mathcal{L}(\theta) = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

BCE

https://stats.stackexchange.com/questions/260505/machine-learning-should-i-use-a-categorical-cross-entropy-or-binary-cross-entro

The math time is over ~~~

- Alarm release….
- The conclusion is that
  - Using BCE is similar to use ML
  - Most often … they are the same
- But …
  can I interpret the weights
  which are given using the machine learning technique?
  - The answer is “it’s not suitable”
    - Because of the relation of the matrix and the samples

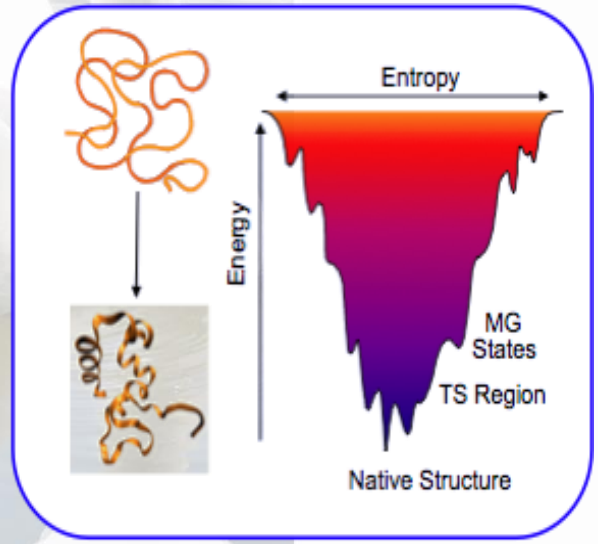
THEN … HOW TO GET THE COEFFICIENT IS INTERESTING

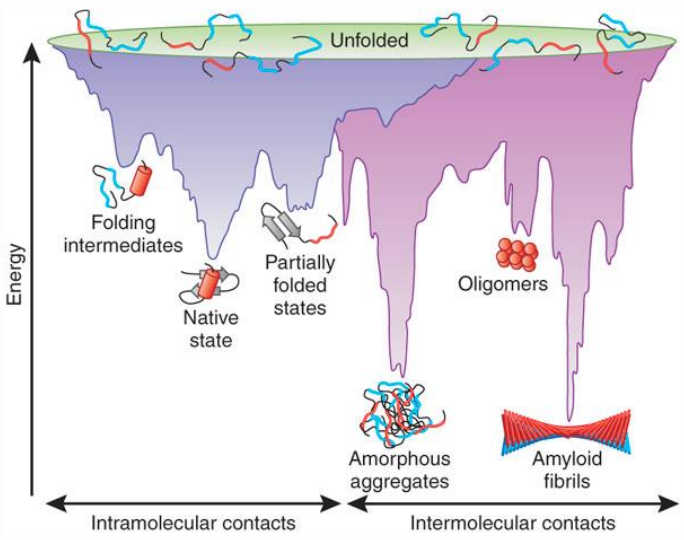
Newton? Gradient Decent?

This time we use particle swarm optimization …

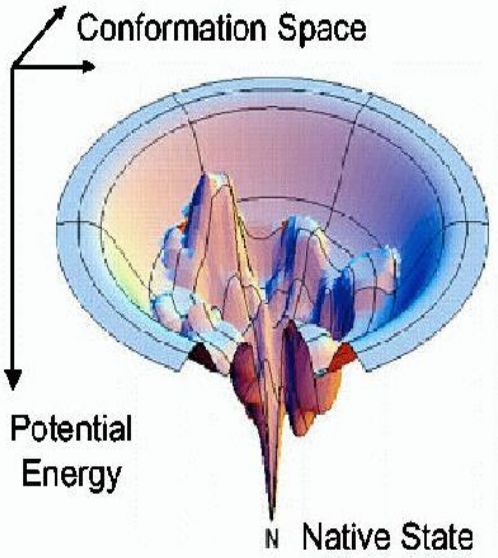
Optimized solution search landscape

Use protein folding as example



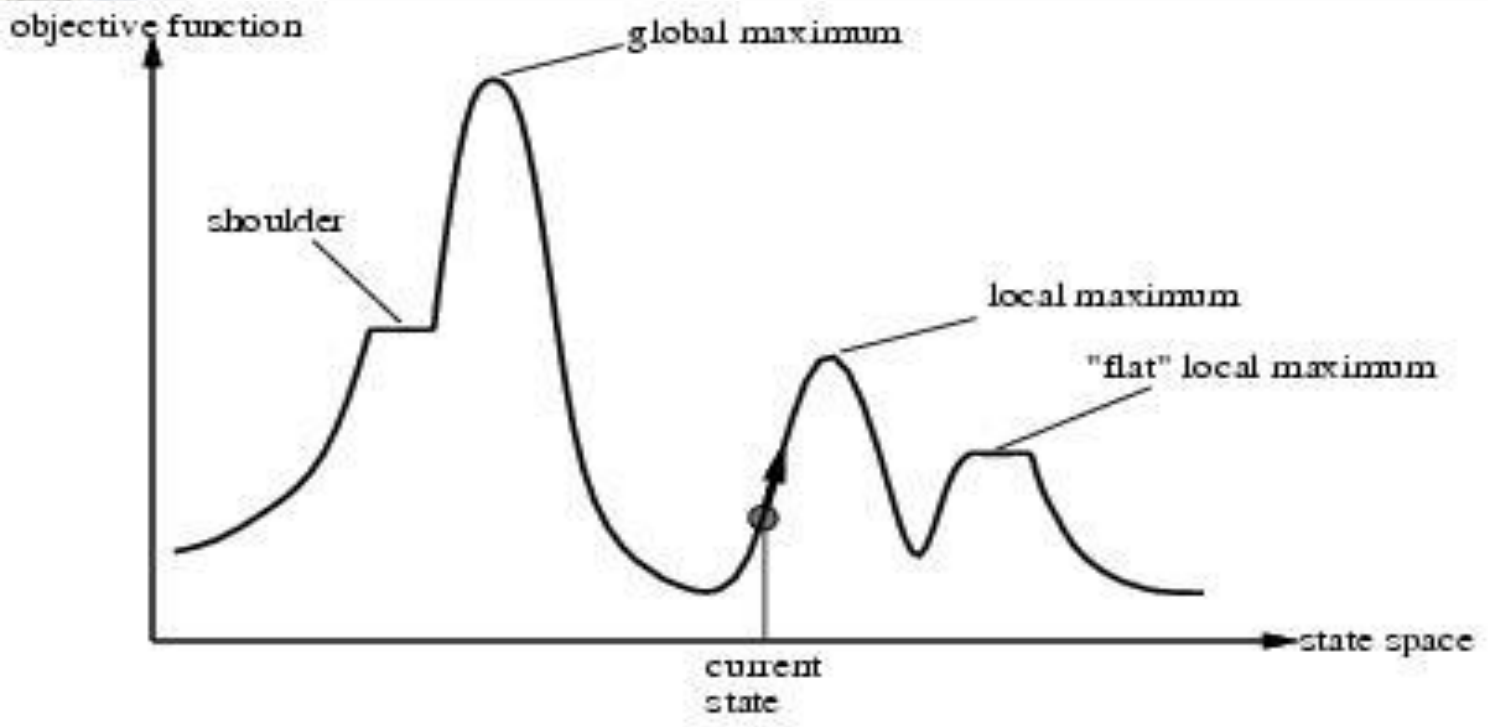


http://www.nature.com/nsmb/journal/v16/n6/fig_tab/nsmb.1591_F1.html



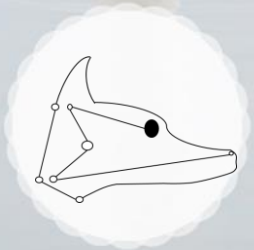
https://parasol.tamu.edu/groups/amatogroup/research/computationalBio/slide/EnergyLandscape.gif

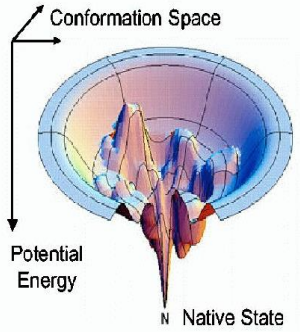
“Landscape” of search



The problems to look for solutions

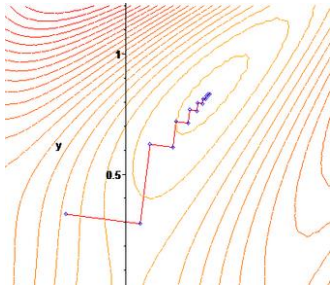
- The only way to get the best solution is to scan all the space.
  - This will take long time.
- If we cannot find the best solution, the acceptable solution would be desired.
  - Traditional method (numerical analysis based)
  - Heuristic algorithm (random based)





Gradient Descent

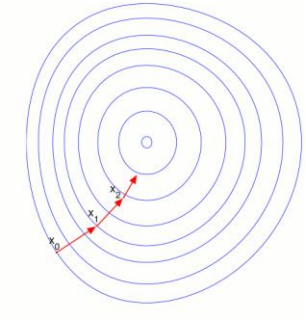
Assume we have some cost-function: $C(x_1,\ldots,x_n)$
and we want minimize over continuous variables X1,X2,..,Xn



1. Compute the *gradient* : $\frac{\partial}{\partial x_i} C(x_1,\ldots,x_n) \qquad \forall i$
2. Take a small step downhill in the direction of the gradient:

$$x_i \rightarrow x'_i = x_i - \lambda \frac{\partial}{\partial x_i} C(x_1,\ldots,x_n) \qquad \forall i$$

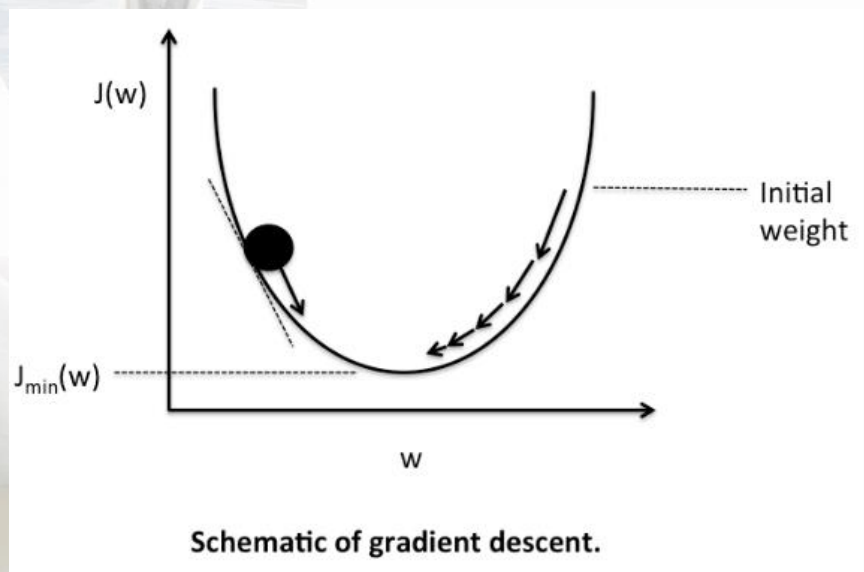
3. Check if $C(x_1,..,x'_i,..,x_n) < C(x_1,..,x_i,..,x_n)$
4. If true then accept move, if not reject.
5. Repeat.



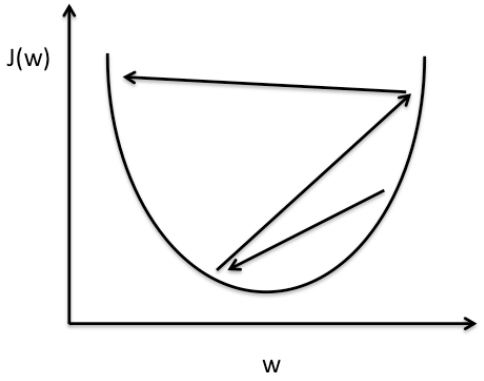
https://www.ics.uci.edu/~welling/teaching/.../LocalSearch271f09.pdf

Problems

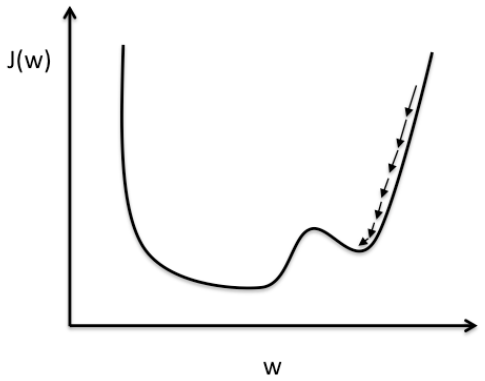
- Gradient decent and learning rate



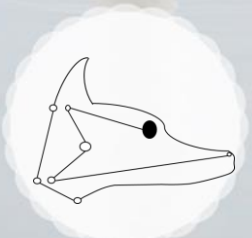
Schematic of gradient descent.



Large learning rate: Overshooting.



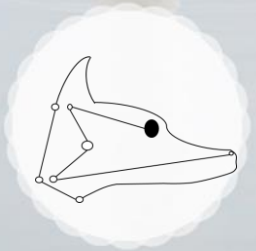
Small learning rate: Many iterations until convergence and trapping in local minima.



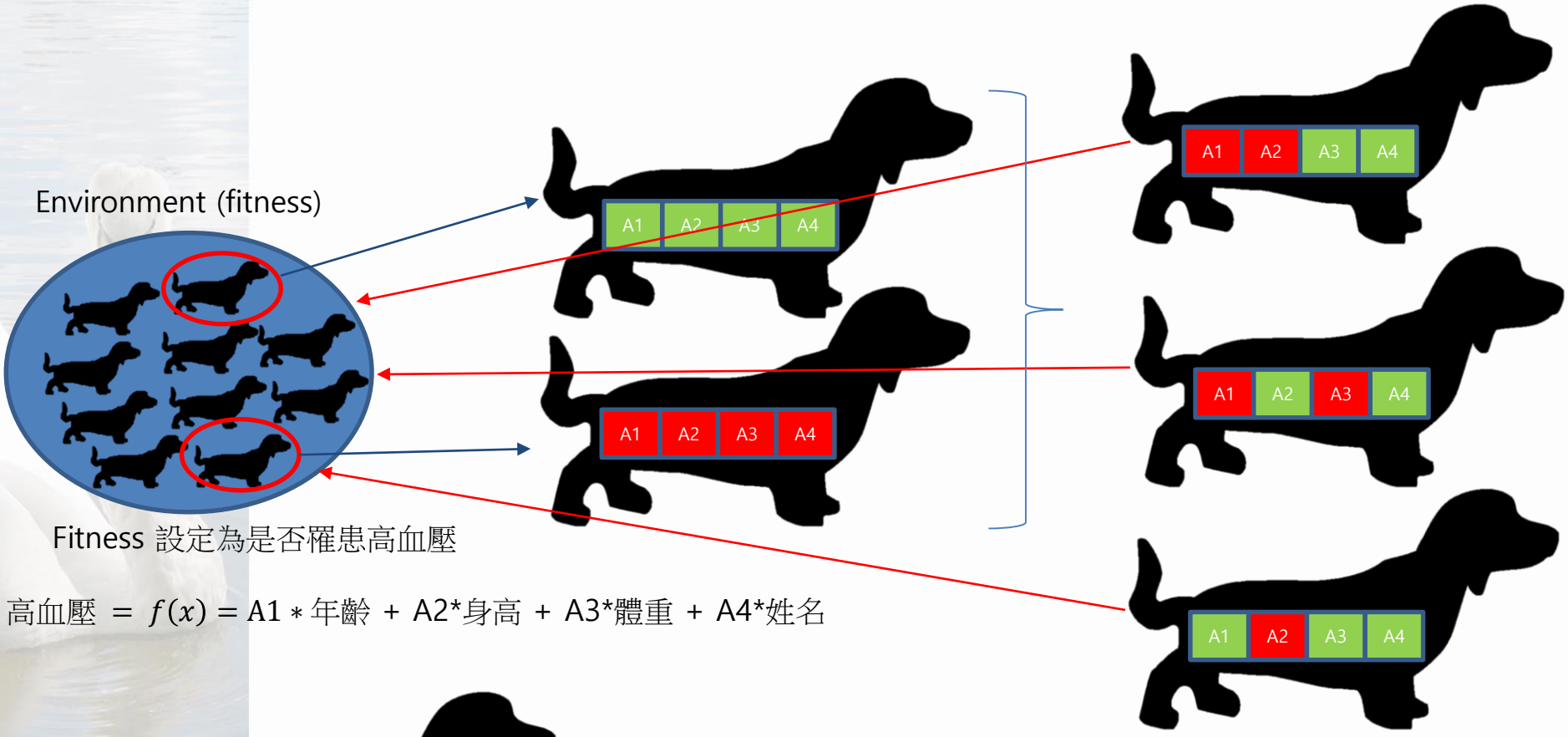
http://sebastianraschka.com/Articles/2015_singlelayer_neurons.html

Why heuristic algorithm

- This kind of method has change to fly over the hill tops.
- Classical and heuristic algorithms are widely apply in many practical areas.
  - Ex: Machine learning



Concepts of GA

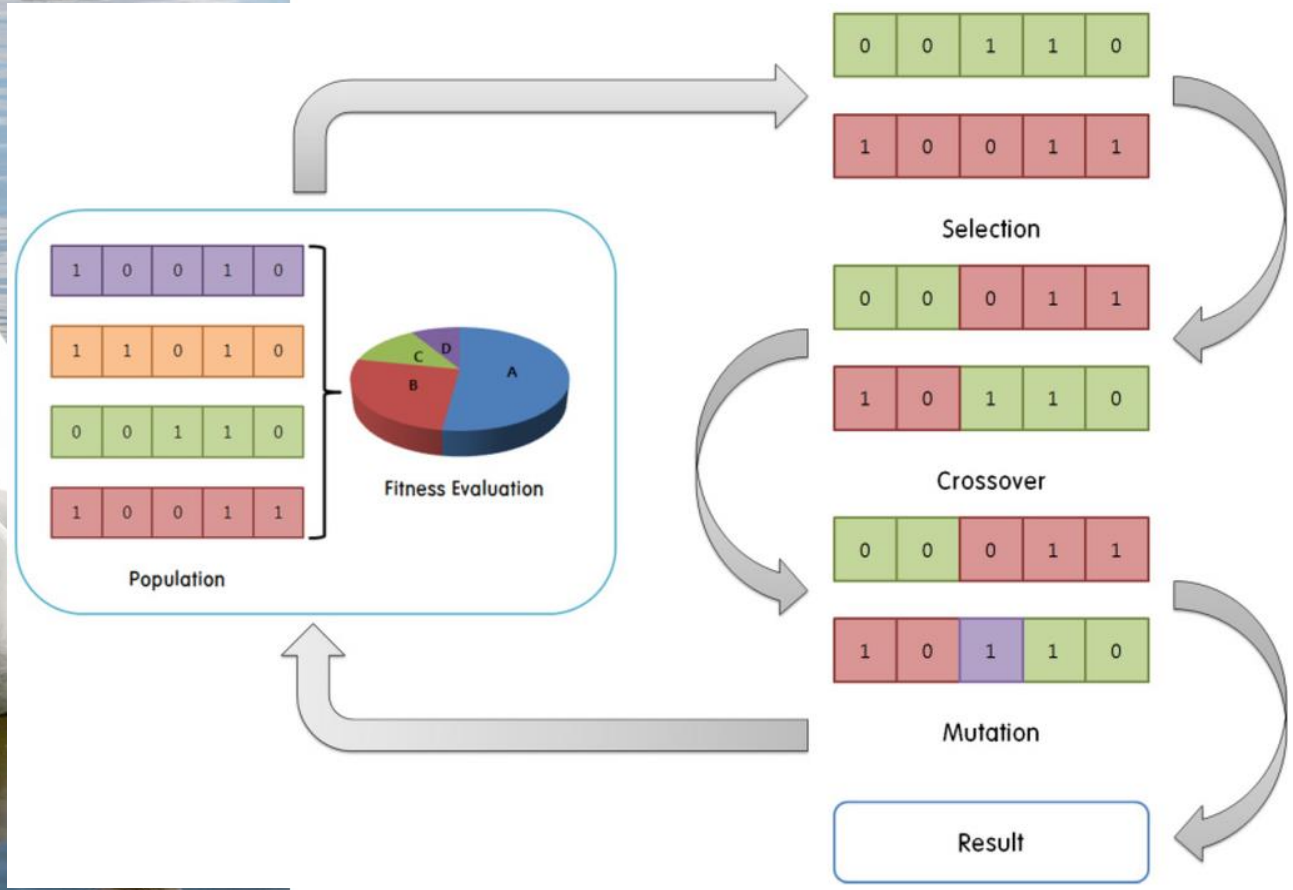


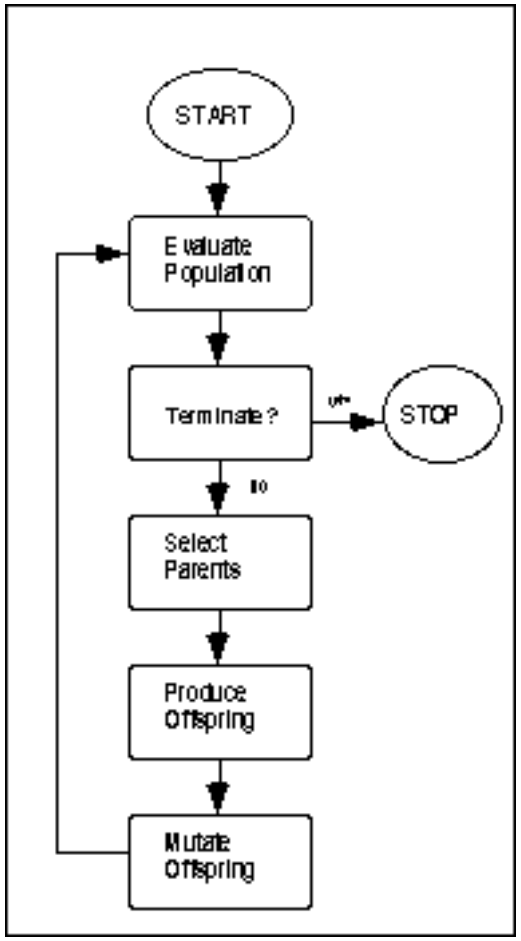
Fitness 設定為是否罹患高血壓

高血壓 $= f(x) =$ A1 * 年齡 + A2*身高 + A3*體重 + A4*姓名

丟回原本的環境後，fitness高的個體就可以存活下來並且有機會把優秀的基因遺傳給下一代

Simple schema of GA

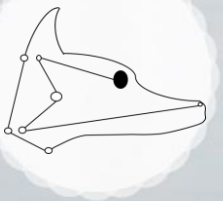




I forget the sources, if anyone know please tell me..

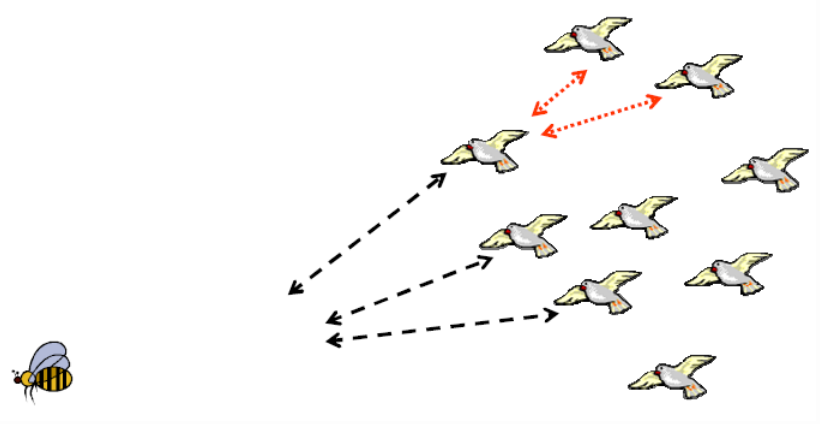
Swarm intelligence

- Nature provides inspiration to computer scientists in many ways. One source of such inspiration is the way in which natural organisms behave
- In other words, if we consider the group itself as an individual – the *swarm* – in some ways, at least, the swarm seems to be more intelligent than any of the individuals within it when they are in groups.
- -- David Corne, Alan Reynolds and Eric Bonabeau, "*Swarm Intelligence* "



Particle swarm optimization (PSO)

- Inference from the birds finding the foods
- All the birds are served as particles in PSO system
- The particles all have some characteristics
  - The memory of current global optima – maybe provided from other birds
  - The memory of current local optima – provided from themselves
  - The velocity of the particle



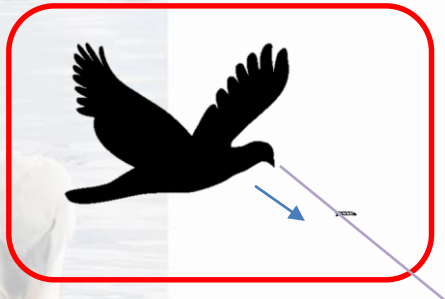
Original target

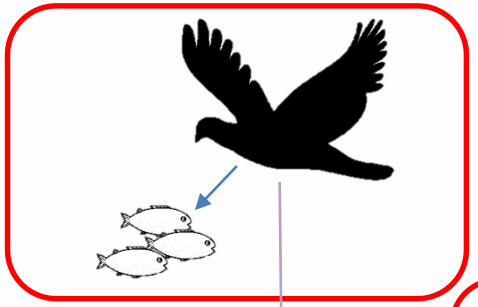
Updated target

Personal optimal solution

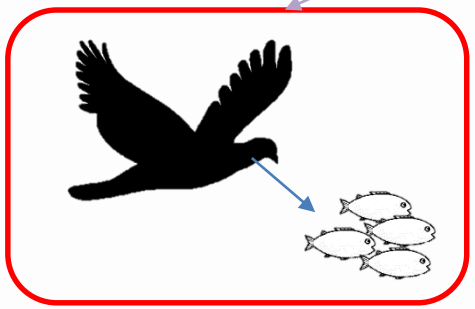
Hypothesis:
The place has more food, the nearby place would have more than more-food

- Boid = Bird – oid (like, mimic etc)
- The birds will move toward to the foods
- But after they passing, the would found another food source.





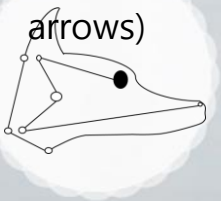
When they know where can get more food, their velocity would be influenced.



Current global optima

All birds will fly here for more food

Considering that:
There are a flock of birds.
All have their own memory of the targets (red rectangles).
All have their velocity to the foods. (blue arrows)



Algorithm – simplified question

- Using the co-ordinates of pbest and gbest, each agent calculates its new velocity as:

$$v_i = v_i + c_1 \times rand() \times (p_{best}x_i - presentx_i) + c_2 \times rand() \times (g_{best}x - presentx_i)$$

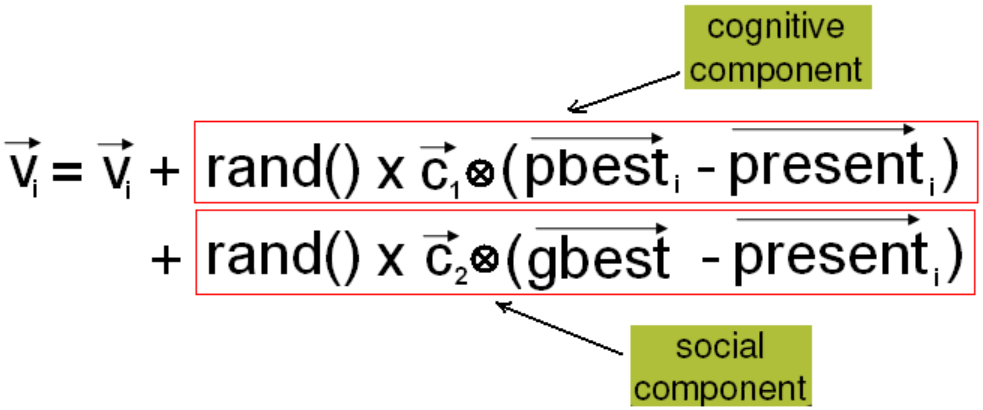
where $0 < rand() < 1$

$$presentx_i = presentx_i + (v_i \times \Delta t)$$

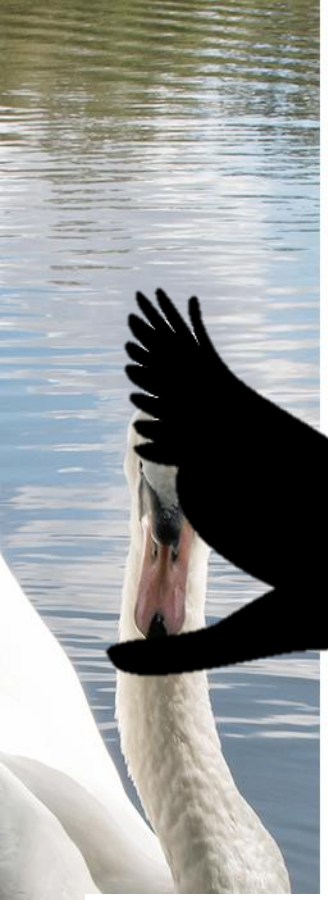
This means all birds will decide they movement according to their own memory and others rewards

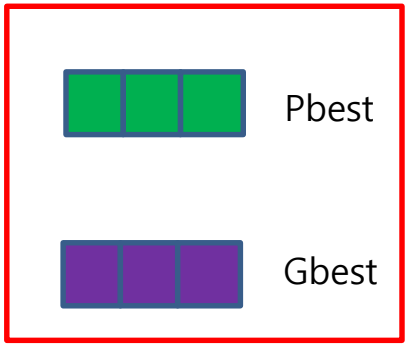
Algorithm – complex question

- In n-dimensional space :

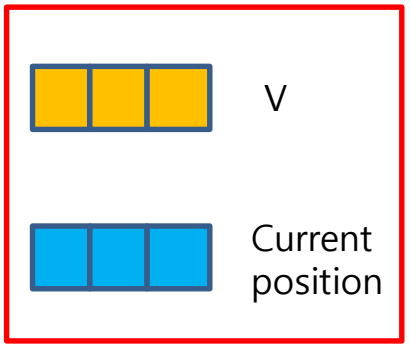


Note that the symbol $\otimes$ denotes a point-wise vector multiplication.

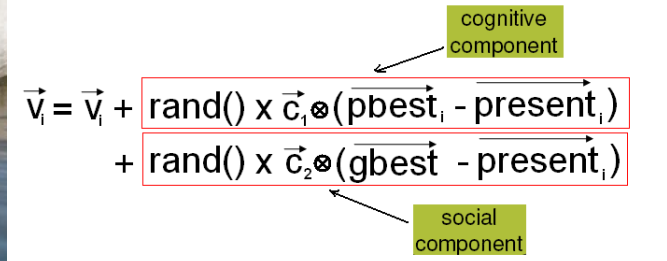




鳥內心對於食物的掙扎



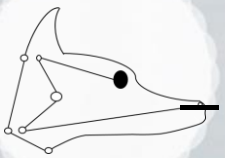
鳥本身的物理狀態



- 乘上一個亂數值，讓鳥不能一次到位(到位了就不用移動了，吃東西就好了)
- 之前覓食就有速度，所以再增加一個慣性給他。(就是不要讓他太快忘記之前曾經經過的地方有多少食物)

Code implementation

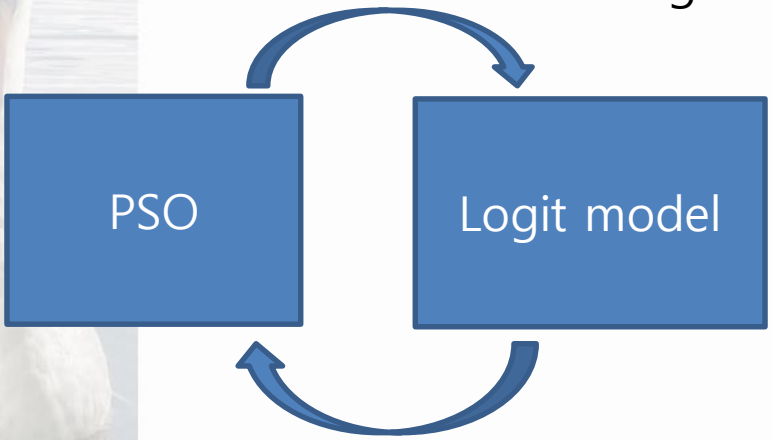
- In this work, we need two basic codes and we need to merge them
  - Logistic regression
    - aymericdamien : TensorFlow-Examples
    - https://goo.gl/LkQxDx
  - PSO
    - Nathan A. Rooy
    - https://goo.gl/89vcSA
  - Dataset
    - Since the original LR use the MNIST, we use the IRIS.csv to instead.
    - Use the in house script to split the data (also available in Github)
  - https://github.com/markliou/LR_PSO_Tensorflow



Some tips in this work

- The flowchart of this script

1. Make the weights and bias tensor
2. Feed this tensor into logit model



- Return the logit model performances (as the fitness)

Tips:

1. Calculating the fitness need to put the tensors in to the session – we need to deliver the session object in to PSO
2. The PSO tensors are made outside the session. Use the implement of "tf.Variables" would acceler ate the speeds

DEMO TIME~~

Any questions ??

THANKS FOR YOUR ATTENTION