Ensemble

團隊合作，一把model一起上！

Framework of Ensemble

有Bagging & Boosting 場合不一樣

- Get a set of classifiers
  - $f_1(x), f_2(x), f_3(x), \ldots\ldots$

坦 補 DD They should be diverse.

整合

- Aggregate the classifiers (*properly*)
  - 在打王時每個人都有該站的位置

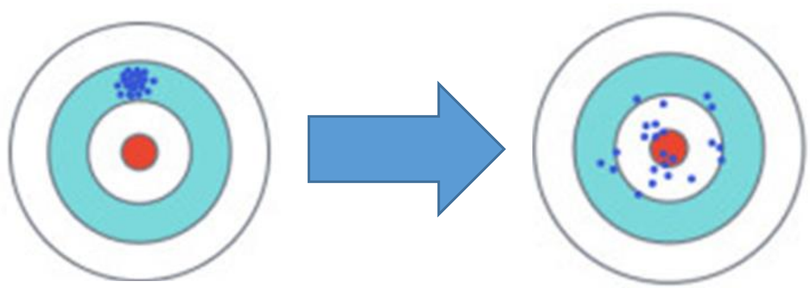
Ensemble: Bagging

Bagging中model的訓練是沒有順序的，可以一次train一坨

Review: Bias v.s. Variance

Large Bias

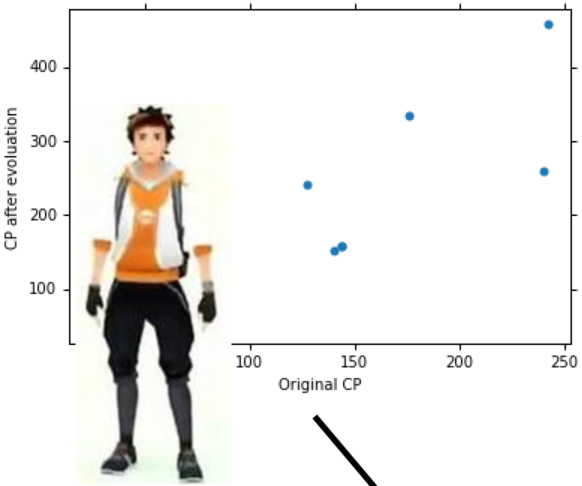
Small Variance



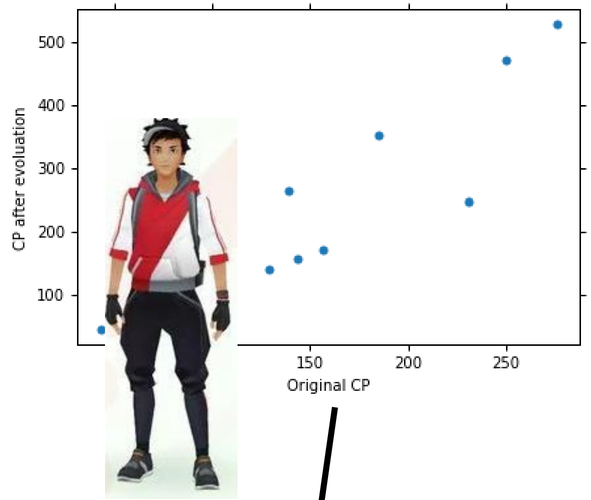
Small Bias

Large Variance

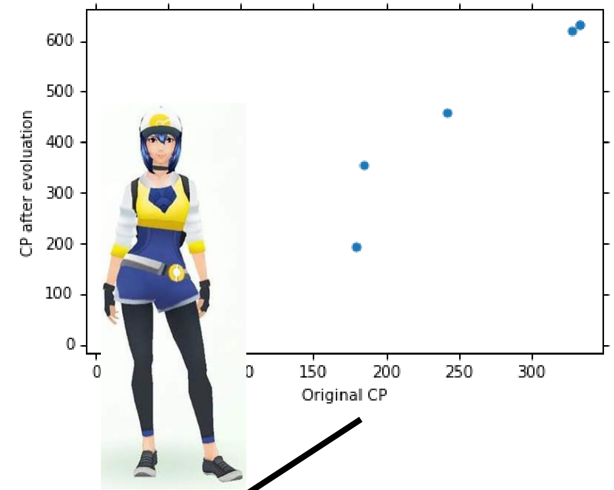
Universe 1



Universe 2

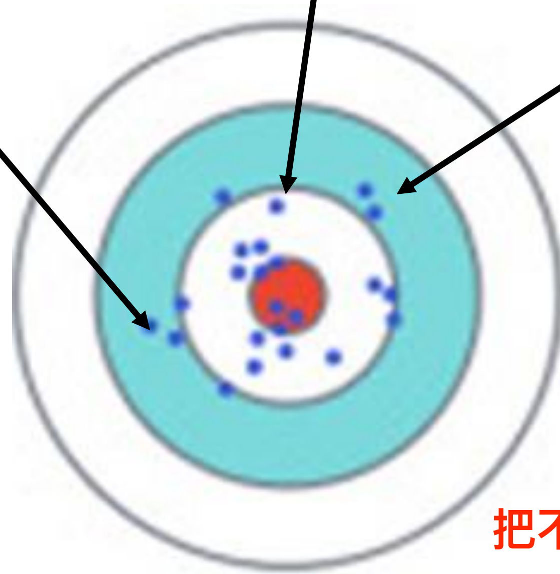


Universe 3



A complex model will have large variance.

We can average complex models to reduce variance.



If we average all the f^*, is it close to $\hat{f}$

$$E[f^*] = \hat{f}$$

把不同的模型的輸出集合起來做平均

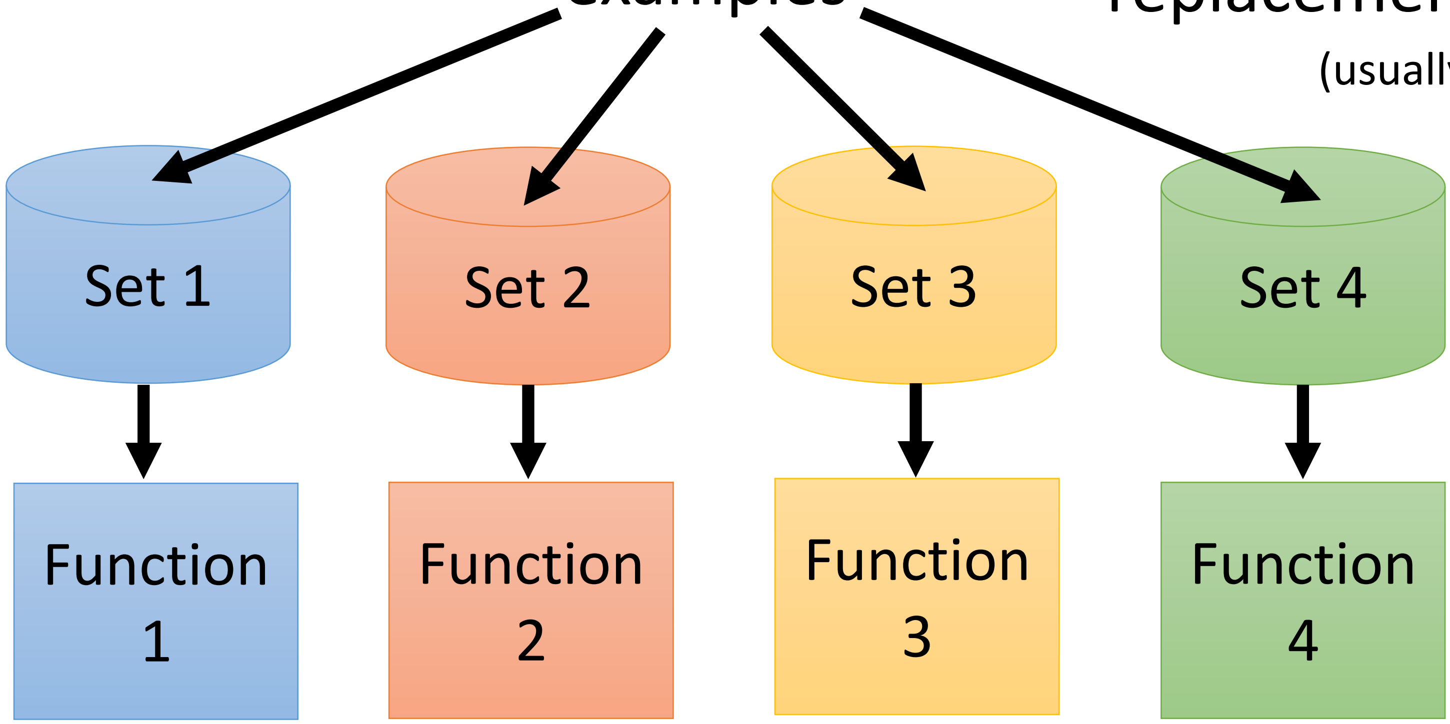
Bagging

sample完通常會放回去

Sampling N’ examples with replacement

(usually N=N’)

N training examples



把training data先分成很多份，在各自針對sub training data去train出一個model

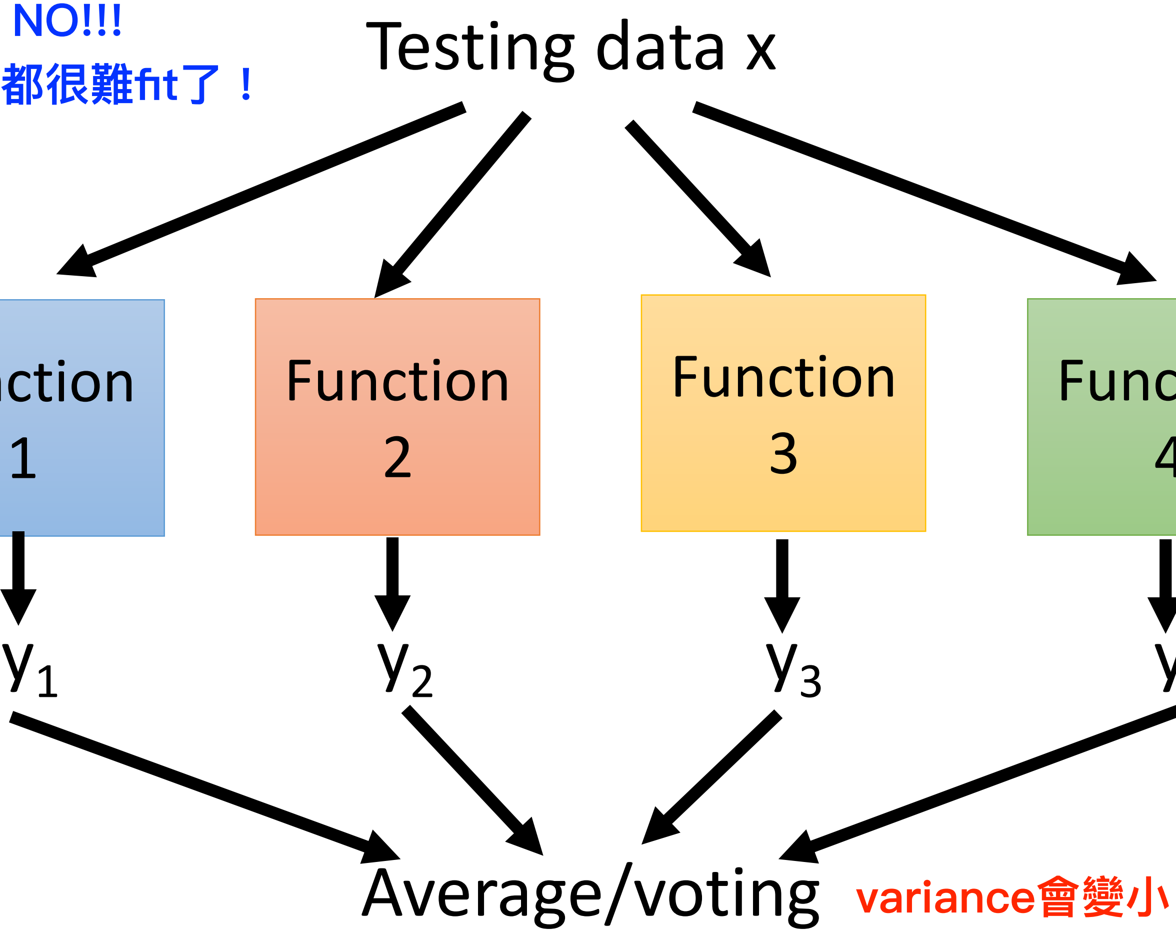
Bagging

This approach would be helpful when your model is complex, easy to overfit.

e.g. decision tree

當model很複雜，擔心他overfitting的時候，為了減低variance才做bagging

NN overfitting!? NO!!!
通常在train data都很難fit了！



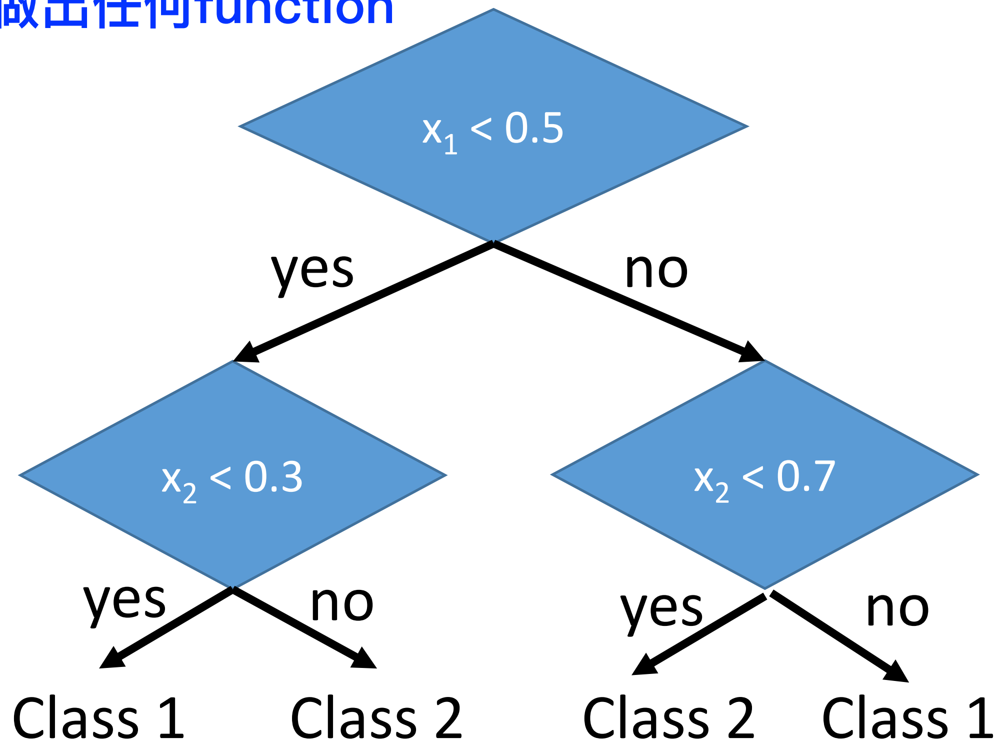
variance會變小

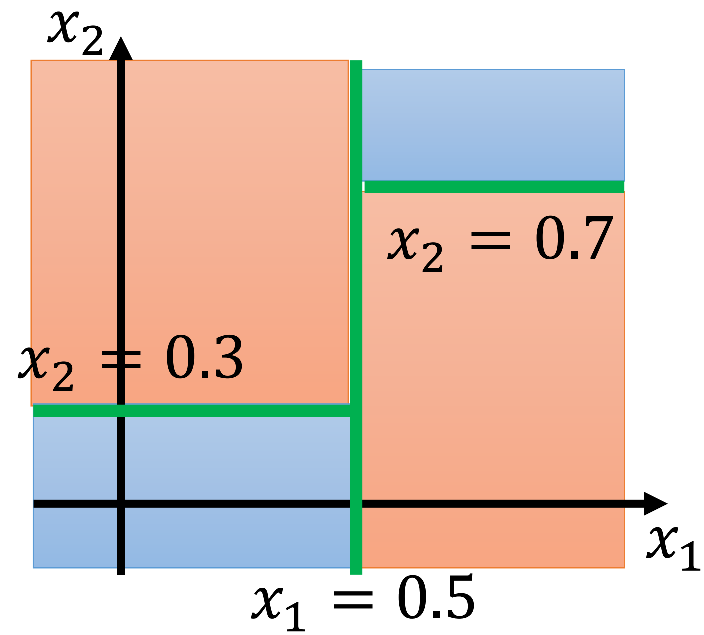
decision tree非常容易overfitting，
因此可做bagging，結果即為
random forest

Decision Tree

樹夠深，decision tree
可做出任何function

Assume each object x is represented by a 2-dim vector $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$





The questions in training .....

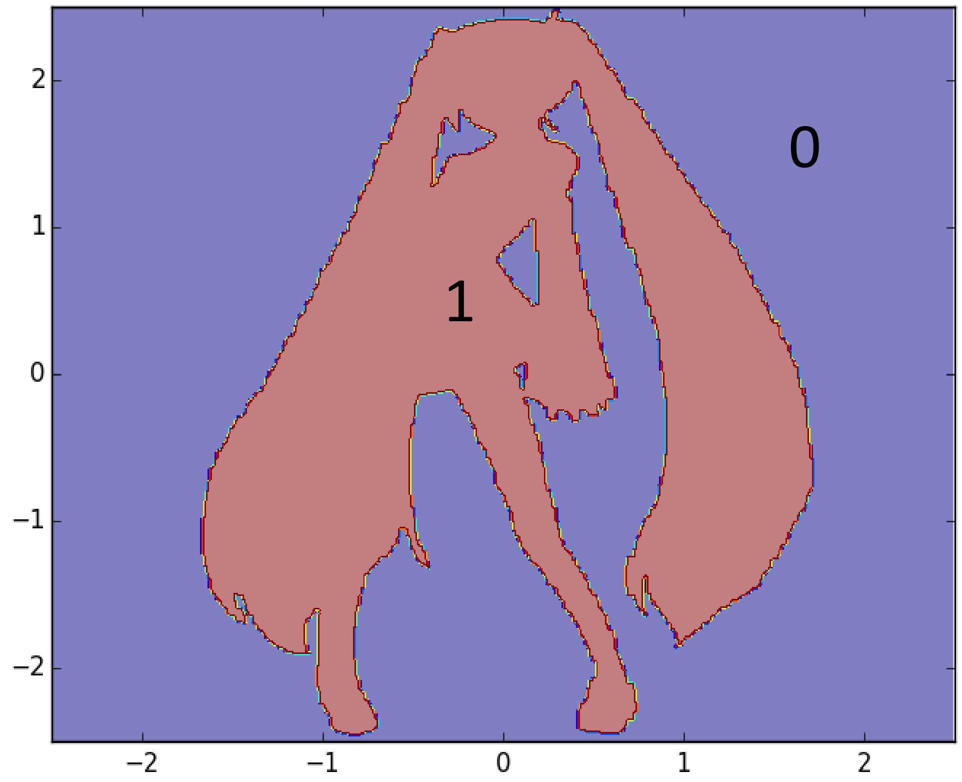
number of branches,
Branching criteria,
termination criteria,
base hypothesis

Can have more complex questions

每個節點要做多少分支，criteria，什麼時候停止分支..等等需要考慮

decision tree example

Experiment: Function of Miku

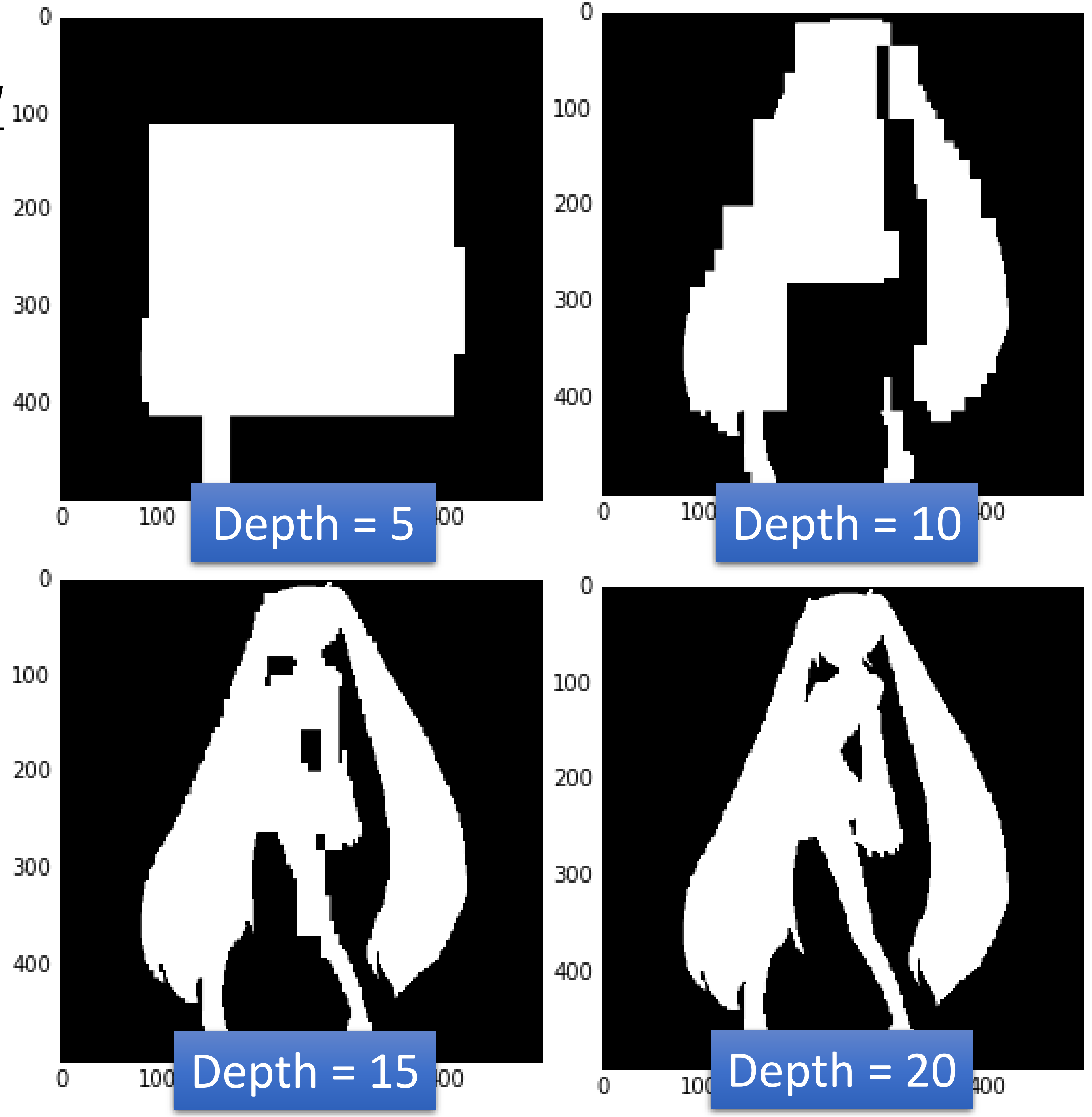


http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2015_2/theano/miku

(1st column: x, 2nd column: y, 3rd column: output (1 or 0))

Experiment: Function of Miku

Single Decision Tree



Random Forest

train	f_1	f_2	f_3	f_4
x^1	O	X	O	X
x^2	O	X	X	O
x^3	X	O	O	X
x^4	X	O	X	O

- Decision tree:
  - Easy to achieve 0% error rate on training data
    - If each training example has its own leaf ……
- Random forest: Bagging of decision tree
  - Resampling training data is not sufficient
  - Randomly restrict the features/questions used in each split random決定哪些feature不能用

特別的sample方法

- Out-of-bag validation for bagging 不需要切validation也可以有validation的效果
  - Using RF = f_2+f_4 to test x^1
  - Using RF = f_2+f_3 to test x^2
  - Using RF = f_1+f_4 to test x^3
  - Using RF = f_1+f_3 to test x^4

Out-of-bag (OOB) error 取平均

Good error estimation of testing set

拿沒看過test data的model去test他的validation accuracy

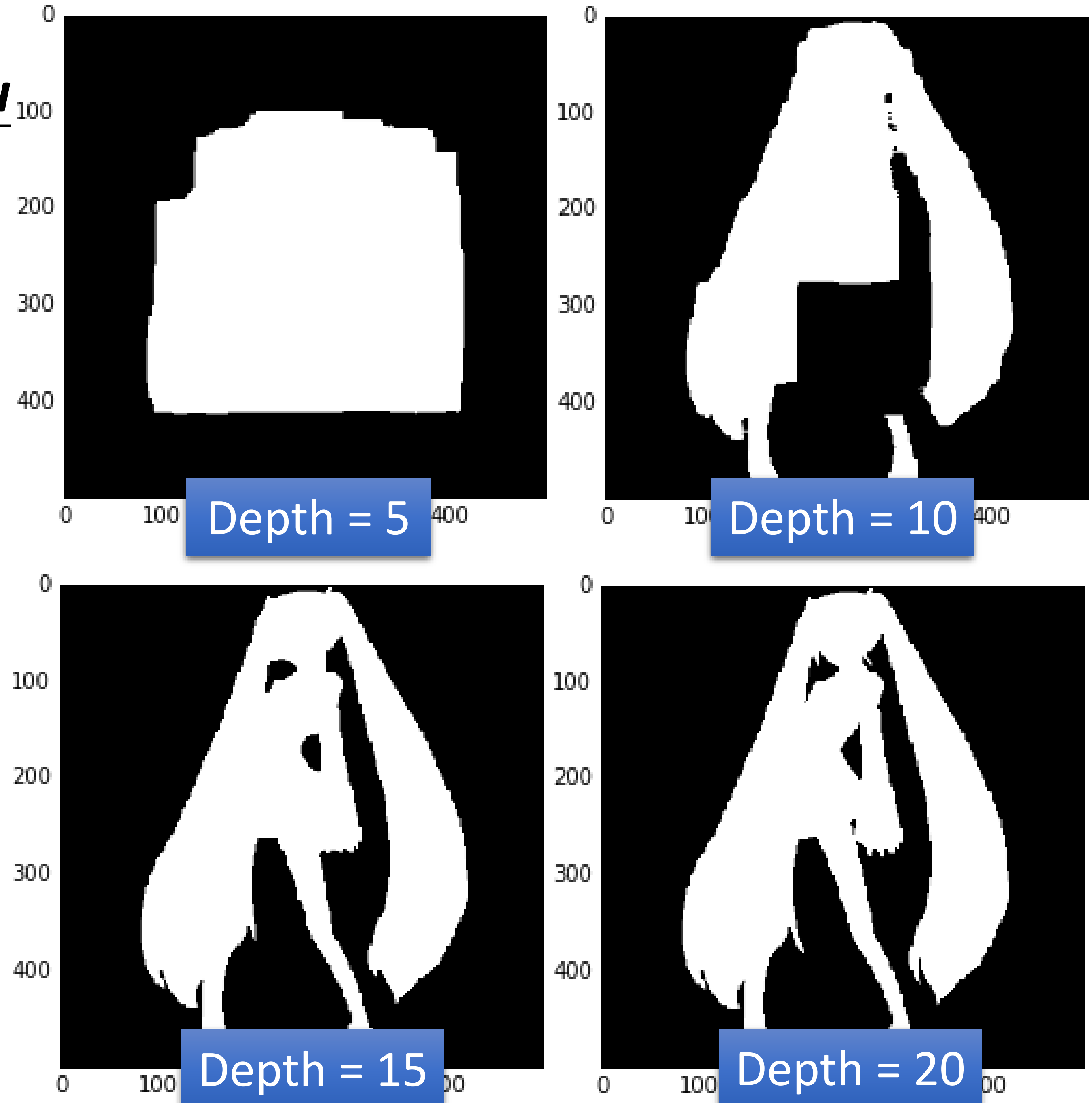
Experiment:

Function of Miku

用random forest並不會讓你更容易去fit data，只是讓結果比較平滑（robust）

Random Forest

(100 trees)



Ensemble: Boosting

Improving Weak Classifiers

bagging是用在很強的model，Boosting是用在很弱的model，當有些raw model但是無法讓他們fit data，這時候可以採用boosting

Boosting

Training data:

$$\{(x^1, \hat{y}^1), \cdots, (x^n, \hat{y}^n), \cdots, (x^N, \hat{y}^N)\}$$

$\hat{y} = \pm 1$ (binary classification)

- Guarantee: 神保證！！
  - If your ML algorithm can produce classifier with error rate smaller than 50% on training data
  - You can obtain 0% error rate classifier after boosting.
- Framework of boosting
  - Obtain the first classifier $f_1(x)$
  - Find another function $f_2(x)$ to help $f_1(x)$
    - However, if $f_2(x)$ is similar to $f_1(x)$, it will not help a lot. f2雖然是輔助f1，但是他們彼此需要是互補的
    - We want $f_2(x)$ to be complementary with $f_1(x)$ (How?)
  - Obtain the second classifier $f_2(x)$
  - …… Finally, combining all the classifiers
- The classifiers are learned sequentially.

model的訓練是有順序的，比須先找出f1，在找出與其互補的f2...以此類推

How to obtain different classifiers?

- Training on different training data sets
- How to have different training data sets
  - Re-sampling your training data to form a new set
  - Re-weighting your training data to form a new set
  - In real implementation, you only have to change the cost/objective function

$(x^1, \hat{y}^1, u^1)$ $u^1 = \cancel{1}$ 0.4

$(x^2, \hat{y}^2, u^2)$ $u^2 = \cancel{1}$ 2.1

$(x^3, \hat{y}^3, u^3)$ $u^3 = \cancel{1}$ 0.7

$$L(f) = \sum_n l(f(x^n), \hat{y}^n)$$

$$\Downarrow$$

乘上權重

$$L(f) = \sum_n \boxed{u^n} l(f(x^n), \hat{y}^n)$$

藉由改變weight來產生不同的data set

sampling也可以想像成改weight，只是weight變成整數（sample到的次數）

boosting的其中一種方法

Idea of Adaboost

training example # : n

step 2. 找一組新的training data在f1上面是會壞掉的

- Idea: **training $f_2(x)$ on the new training set that fails $f_1(x)$**
- How to find a new training set that fails $f_1(x)$?

ε_1: the error rate of $f_1(x)$ on its training data

weight

$$\varepsilon_1 = \frac{\sum_n u_1^n \delta(f_1(x^n) \neq \hat{y}^n)}{Z_1} \qquad Z_1 = \sum_n u_1^n \qquad \varepsilon_1 < 0.5$$

normalization

假設錯誤率小於五十趴

step 2.

Changing the example weights from u_1^n to u_2^n such that

$$\frac{\sum_n u_2^n \delta(f_1(x^n) \neq \hat{y}^n)}{Z_2} = 0.5$$

使其壞掉

The performance of f_1 for new weights would be random.

Training $f_2(x)$ based on the new weights u_2^n

Re-weighting Training Data

- Idea: **training $\boldsymbol{f_2(x)}$ on the new training set that fails $\boldsymbol{f_1(x)}$**
- How to find a new training set that fails $f_1(x)$?

把答對的題目權重變小，把答錯的題目權重變大X D這樣就可以把f1的結果搞爛找出一組與其互補的new data set

$(x^1, \hat{y}^1, u^1)$ $u^1 = 1$ $\quad\quad$ $u^1 = 1/\sqrt{3}$

$(x^2, \hat{y}^2, u^2)$ $u^2 = 1$ $\quad\quad$ $u^2 = \sqrt{3}$

$(x^3, \hat{y}^3, u^3)$ $u^3 = 1$ $\quad\quad$ $u^3 = 1/\sqrt{3}$

$(x^4, \hat{y}^4, u^4)$ $u^4 = 1$ $\quad\quad$ $u^4 = 1/\sqrt{3}$

ε_1 =0.25

1/4

$f_1(x)$

0.5

ε_2 <0.5

$f_2(x)$

拿新的data set去train出f2，訓練出error rate<0.5

Re-weighting Training Data

- Idea: **training $\boldsymbol{f_2(x)}$ on the new training set that fails $\boldsymbol{f_1(x)}$**
- How to find a new training set that fails $f_1(x)$?

If x^n misclassified by f_1 ($f_1(x^n) \neq \hat{y}^n$)

$u_2^n \leftarrow u_1^n$ multiplying d_1 increase

If x^n correctly classified by f_1 ($f_1(x^n) = \hat{y}^n$)

$u_2^n \leftarrow u_1^n$ divided by d_1 decrease

f_2 will be learned based on example weights u_2^n

What is the value of d_1?

Re-weighting Training Data

$$\varepsilon_1 = \frac{\sum_n u_1^n \delta(f_1(x^n) \neq \hat{y}^n)}{\boxed{Z_1}} \qquad Z_1 = \sum_n u_1^n$$

分類錯誤的地方

$$\frac{\sum_n u_2^n \delta(f_1(x^n) \neq \hat{y}^n)}{\boxed{Z_2}} = 0.5$$

$f_1(x^n) \neq \hat{y}^n$ $\quad u_2^n \leftarrow u_1^n$ multiplying d_1

$f_1(x^n) = \hat{y}^n$ $\quad u_2^n \leftarrow u_1^n$ divided by d_1

$$= \sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1 \qquad = \sum_{f_1(x^n) \neq \hat{y}^n} u_2^n + \sum_{f_1(x^n) = \hat{y}^n} u_2^n$$

$$= \sum_n u_2^n \qquad = \sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1 + \sum_{f_1(x^n) = \hat{y}^n} u_1^n / d_1$$

分類錯誤的地方 分類正確的地方

$$\frac{\sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1 + \sum_{f_1(x^n) = \hat{y}^n} u_1^n / d_1}{\sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1} = 2$$

分類錯誤的地方

Re-weighting Training Data

epsilon <= 0.5

$$\varepsilon_1 = \frac{\sum_n u_1^n \delta(f_1(x^n) \neq \hat{y}^n)}{Z_1} \qquad Z_1 = \sum_n u_1^n$$

$$\frac{\sum_n u_2^n \delta(f_1(x^n) \neq \hat{y}^n)}{Z_2} = 0.5$$

$f_1(x^n) \neq \hat{y}^n$ $u_2^n \leftarrow u_1^n$ multiplying d_1

$f_1(x^n) = \hat{y}^n$ $u_2^n \leftarrow u_1^n$ divided by d_1

$$\frac{\sum_{f_1(x^n)\neq\hat{y}^n} u_1^n d_1 + \sum_{f_1(x^n)=\hat{y}^n} u_1^n / d_1}{\sum_{f_1(x^n)\neq\hat{y}^n} u_1^n d_1} = 2 \qquad \frac{\sum_{f_1(x^n)=\hat{y}^n} u_1^n / d_1}{\sum_{f_1(x^n)\neq\hat{y}^n} u_1^n d_1} = 1$$

$$\sum_{f_1(x^n)=\hat{y}^n} u_1^n / d_1 = \sum_{f_1(x^n)\neq\hat{y}^n} u_1^n d_1$$

$$\boxed{\frac{1}{d_1} \underbrace{\sum_{f_1(x^n)=\hat{y}^n} u_1^n}_{Z_1(1-\varepsilon_1)} = d_1 \underbrace{\sum_{f_1(x^n)\neq\hat{y}^n} u_1^n}_{Z_1\varepsilon_1}}$$

$$\varepsilon_1 = \frac{\sum_{f_1(x^n)\neq\hat{y}^n} u_1^n}{Z_1}$$

答錯的總和

$$\sum_{f_1(x^n)\neq\hat{y}^n} u_1^n = Z_1\varepsilon_1$$

$$Z_1(1-\varepsilon_1)/d_1 = Z_1\varepsilon_1 d_1$$

$$\boxed{d_1 = \sqrt{(1-\varepsilon_1)/\varepsilon_1} > 1}$$

結論

Algorithm for AdaBoost

- Giving training data
  $\{(x^1, \hat{y}^1, u_1^1), \cdots, (x^n, \hat{y}^n, u_1^n), \cdots, (x^N, \hat{y}^N, u_1^N)\}$
  - $\hat{y} = \pm 1$ (Binary classification), $u_1^n = 1$ (equal weights)
- For t = 1, …, T:
  - Training weak classifier $f_t(x)$ with weights $\{u_t^1, \cdots, u_t^N\}$
  - ε_t is the error rate of $f_t(x)$ with weights $\{u_t^1, \cdots, u_t^N\}$
  - For n = 1, …, N:
    - If x^n is misclassified by $f_t(x)$: $\hat{y}^n \neq f_t(x^n)$
    - $u_{t+1}^n = u_t^n \times d_t = u_t^n \times \exp(\alpha_t)$
    - Else:
    - $u_{t+1}^n = u_t^n / d_t = u_t^n \times \exp(-\alpha_t)$

分類錯誤

分類正確

$$d_t = \sqrt{(1-\varepsilon_t)/\varepsilon_t}$$

$$\alpha_t = ln\sqrt{(1-\varepsilon_t)/\varepsilon_t}$$

$$u_{t+1}^n \leftarrow u_t^n \times exp(\quad -y\text{^}n * ft(xn) \quad \alpha_t)$$

換成exponential是為了將式子表達簡略

Algorithm for AdaBoost

- We obtain a set of functions: $f_1(x), \ldots, f_t(x), \ldots, f_T(x)$
- How to aggregate them?
  - Uniform weight:
    - $H(x) = sign(\sum_{t=1}^{T} f_t(x))$
  - Non-uniform weight:
    - $H(x) = sign(\sum_{t=1}^{T} \alpha_t f_t(x))$

Smaller error ε_t, larger weight for final voting

因為每個classification有不同的好壞之分，因此需要在前面先呈上一個權重（alpha t）

$$\alpha_t = ln\sqrt{(1-\varepsilon_t)/\varepsilon_t}$$

$$u_{t+1}^n = u_t^n \times exp(-\hat{y}^n f_t(x^n)\alpha_t)$$

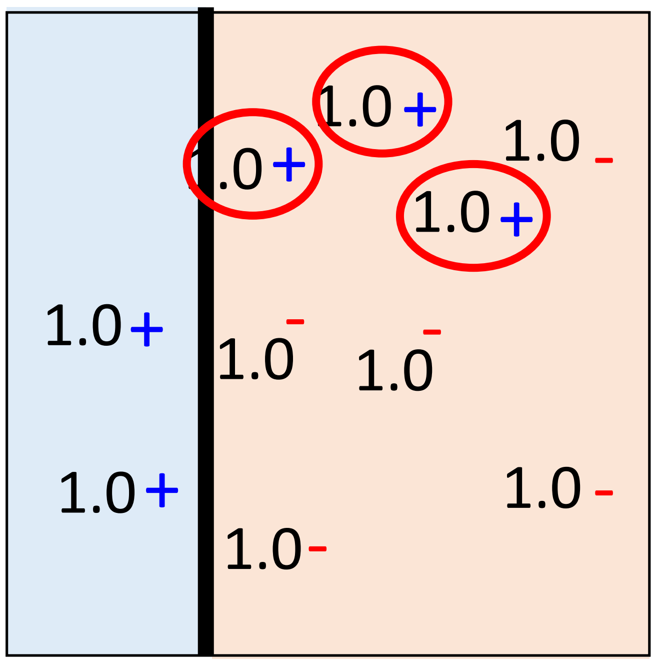
$\varepsilon_t = 0.1$	$\varepsilon_t = 0.4$
α_t =1.10	α_t =0.20

Toy Example

假設分佈在二維平面上，切一刀分類（超弱）

T=3, weak classifier = decision stump

- t=1

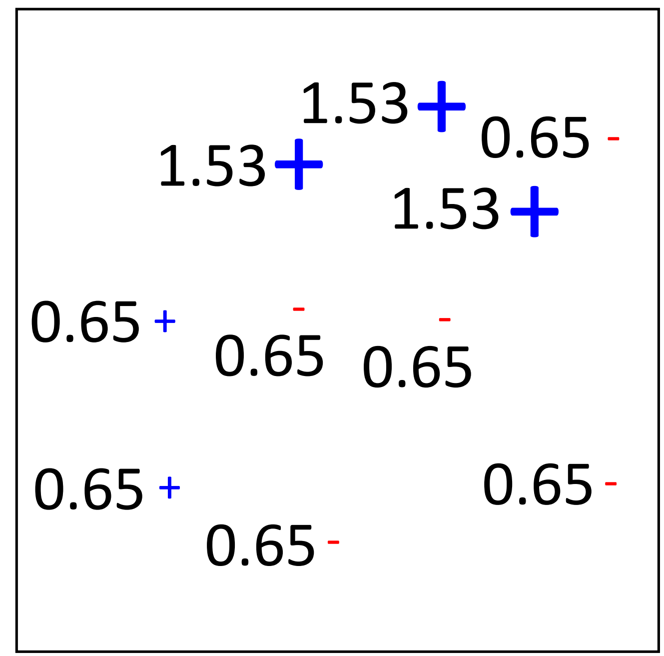


$f_1(x)$

$\varepsilon_1 = 0.30$

$d_1 = 1.53$

$\alpha_1 = 0.42$



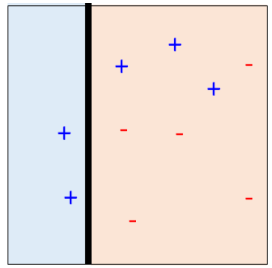
Toy Example

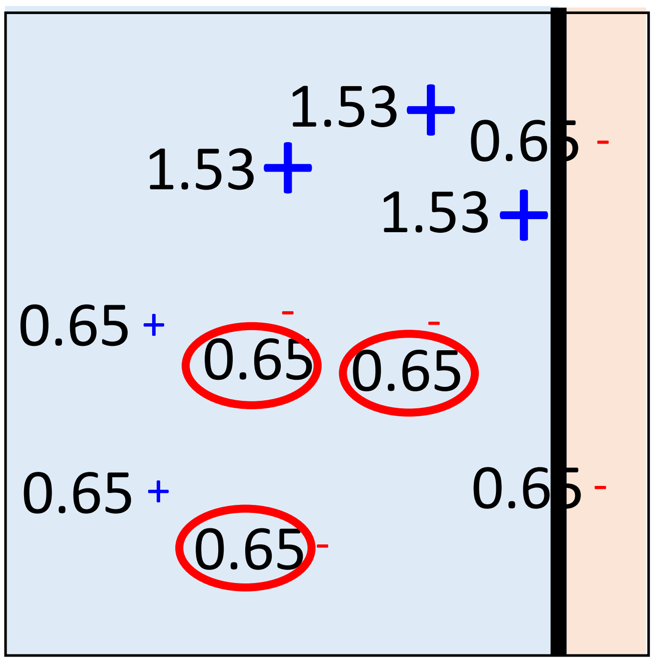
T=3, weak classifier = decision stump

- t=2

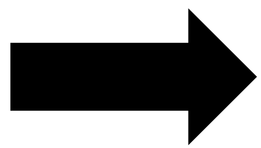
$f_1(x)$:

$\alpha_1 = 0.42$





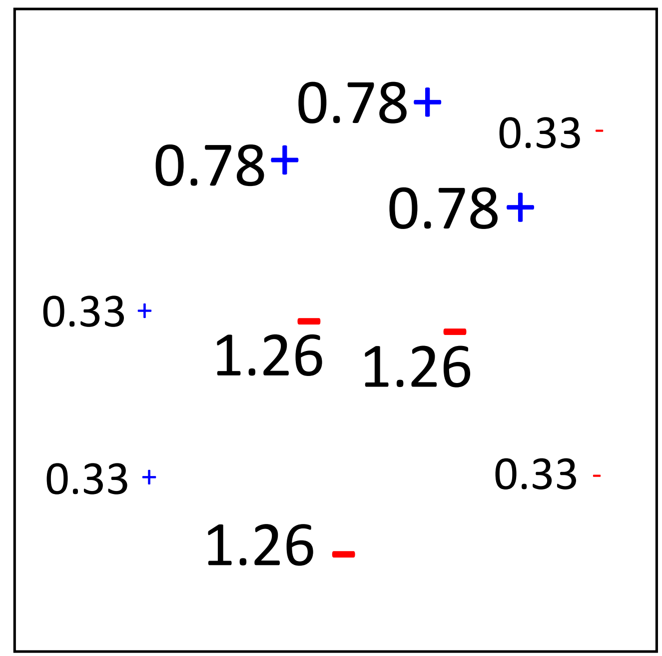
$f_2(x)$



$\varepsilon_2 = 0.21$

$d_2 = 1.94$

$\alpha_2 = 0.66$



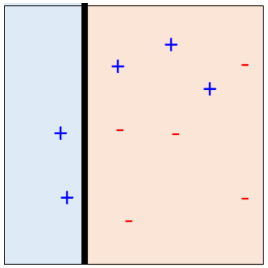
Toy Example

T=3, weak classifier = decision stump

- t=3

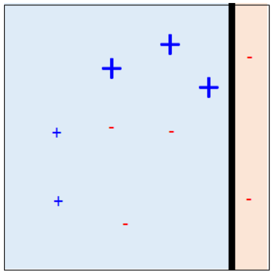
$f_1(x)$:

$\alpha_1 = 0.42$



$f_2(x)$:

$\alpha_2 = 0.66$



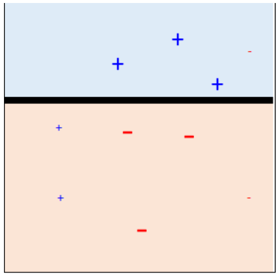
$\varepsilon_3 = 0.13$

$d_3 = 2.59$

$\alpha_3 = 0.95$

$f_3(x)$:

$\alpha_3 = 0.95$

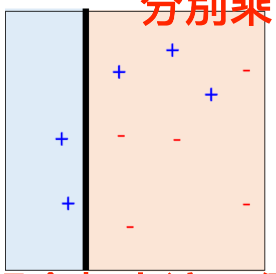
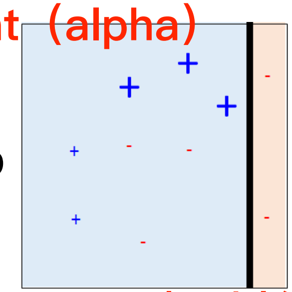
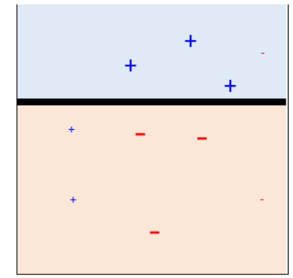


Toy Example

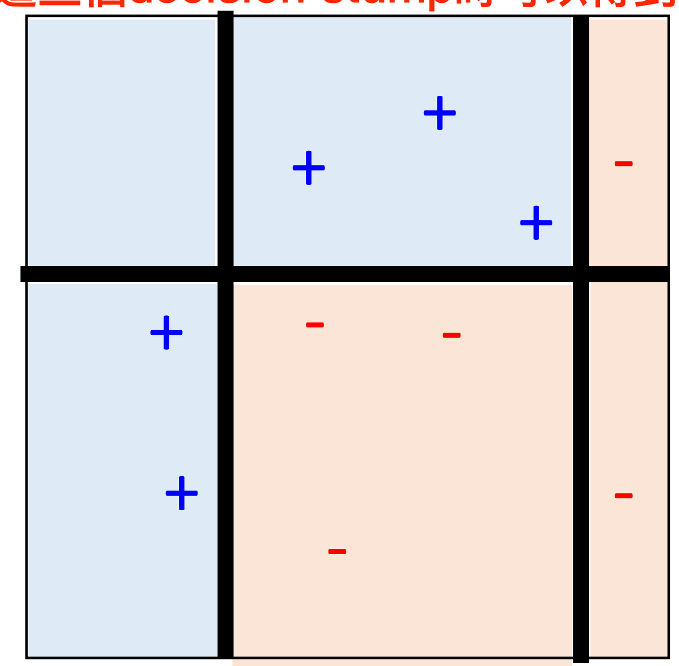
全部加起來看正負

- Final Classifier: $H(x) = sign(\sum_{t=1}^{T} \alpha_t f_t(x))$

分別乘上其weight（alpha)

$sign(\ 0.42$  $+\ 0.66$  $+\ 0.95$  $)$

當組合起來這三個decision stump時可以得到正確率100的結果



Warning of Math

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t f_t(x)\right)$$

epsilon t: classifier ft 的error rate

$$\alpha_t = ln\sqrt{(1-\varepsilon_t)/\varepsilon_t}$$

As we have more and more f_t (T increases), $H(x)$ achieves smaller and smaller error rate on training data.

隨著iteration越高adaboost結果越好

Error Rate of Final Classifier

- Final classifier: $H(x) = sign(\underbrace{\sum_{t=1}^{T} \alpha_t f_t(x)}_{g(x)})$
  - $\alpha_t = ln\sqrt{(1-\varepsilon_t)/\varepsilon_t}$

Training Data Error Rate

$$= \frac{1}{N}\sum_n \delta(H(x^n) \neq \hat{y}^n)$$

判斷同號異號

$$= \frac{1}{N}\sum_n \underline{\delta(\hat{y}^n g(x^n) < 0)}$$

$$\leq \frac{1}{N}\sum_n \underline{exp(-\hat{y}^n g(x^n))}$$

upper bound

$\hat{y}^n g(x^n)$

證明upper bound會越來越小

Training Data Error Rate

$$\leq \frac{1}{N}\sum_n exp(-\hat{y}^n g(x^n)\ ) \boxed{= \frac{1}{N} Z_{T+1}}$$

$$g(x) = \sum_{t=1}^{T} \alpha_t f_t(x)$$

$$\alpha_t = ln\sqrt{(1-\varepsilon_t)/\varepsilon_t}$$

Z_t: the summation of the weights of training data for training f_t

What is $Z_{T+1} =?$ $\quad Z_{T+1} = \sum_n u_{T+1}^n$

第1個iteration

$$\left.\begin{aligned} & u_1^n = 1 \\ & u_{t+1}^n = u_t^n \times exp(-\hat{y}^n f_t(x^n)\alpha_t) \end{aligned}\right\} \quad u_{T+1}^n = \prod_{t=1}^{T} exp(-\hat{y}^n f_t(x^n)\alpha_t)$$

控制增加或是減少weight

$$Z_{T+1} = \sum_n \prod_{t=1}^{T} exp(-\hat{y}^n f_t(x^n)\alpha_t)$$

y^：label data固定的，可以提出

$$= \sum_n exp\left(-\hat{y}^n \overset{g(x)}{\underline{\sum_{t=1}^{T} f_t(x^n)\alpha_t}}\right)$$

只要證明train data的weight之sum越來越小即可得證

Training Data Error Rate

$$\leq \frac{1}{N}\sum_n exp(-\hat{y}^n g(x^n)) = \frac{1}{N}Z_{T+1}$$

$$g(x) = \sum_{t=1}^{T} \alpha_t f_t(x)$$

$$\alpha_t = ln\sqrt{(1-\varepsilon_t)/\varepsilon_t}$$

$Z_1 = N$ (equal weights)

$$Z_t = \underline{Z_{t-1}\varepsilon_t} exp(\alpha_t) + \underline{Z_{t-1}(1-\varepsilon_t)} exp(-\alpha_t)$$

$\boxed{\text{Misclassified portion in } Z_{t-1}}$ $\boxed{\text{Correctly classified portion in } Z_{t-1}}$

$$= Z_{t-1}\varepsilon_t\sqrt{(1-\varepsilon_t)/\varepsilon_t} + Z_{t-1}(1-\varepsilon_t)\sqrt{\varepsilon_t/(1-\varepsilon_t)}$$

$$= Z_{t-1} \times 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$$

$$Z_{T+1} = N\prod_{t=1}^{T} 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$$

$$\boxed{\text{Training Data Error Rate} \leq \prod_{t=1}^{T} \underline{2\sqrt{\epsilon_t(1-\epsilon_t)}}}$$

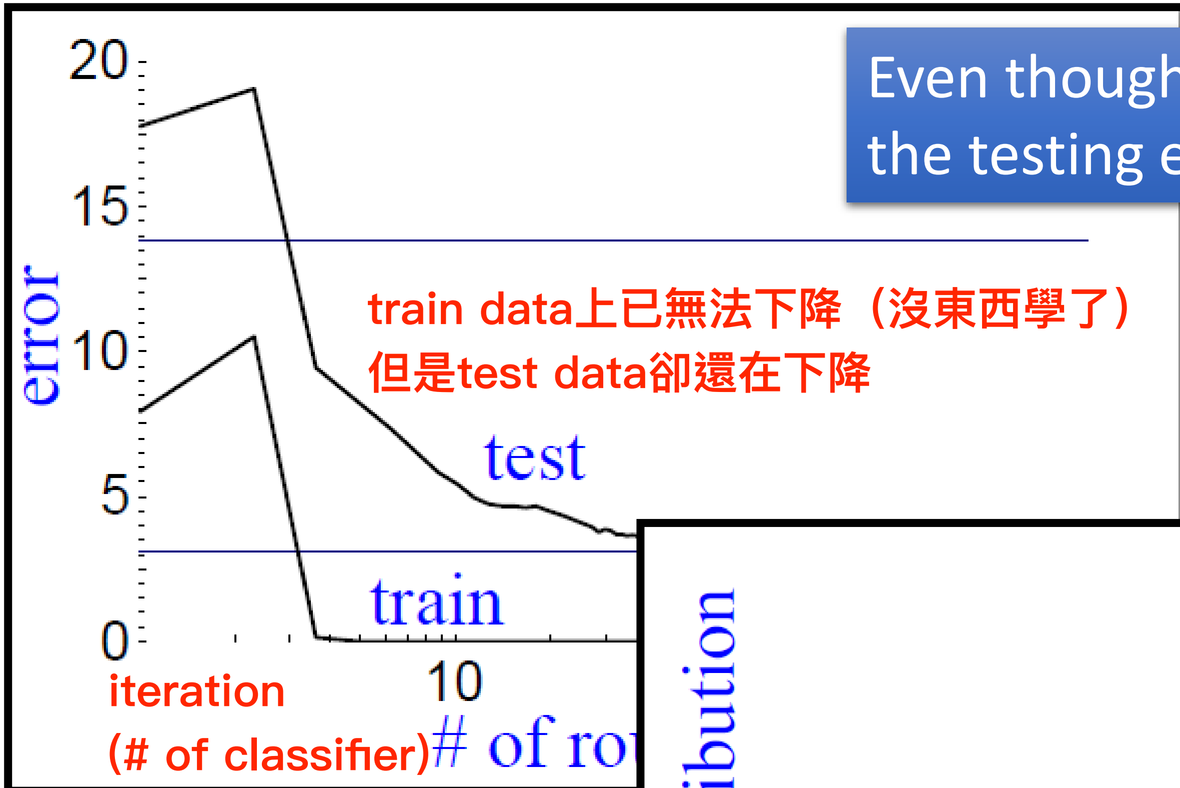
<1

Smaller and smaller

得證

End of Warning

Even though the training error is 0, the testing error still decreases?

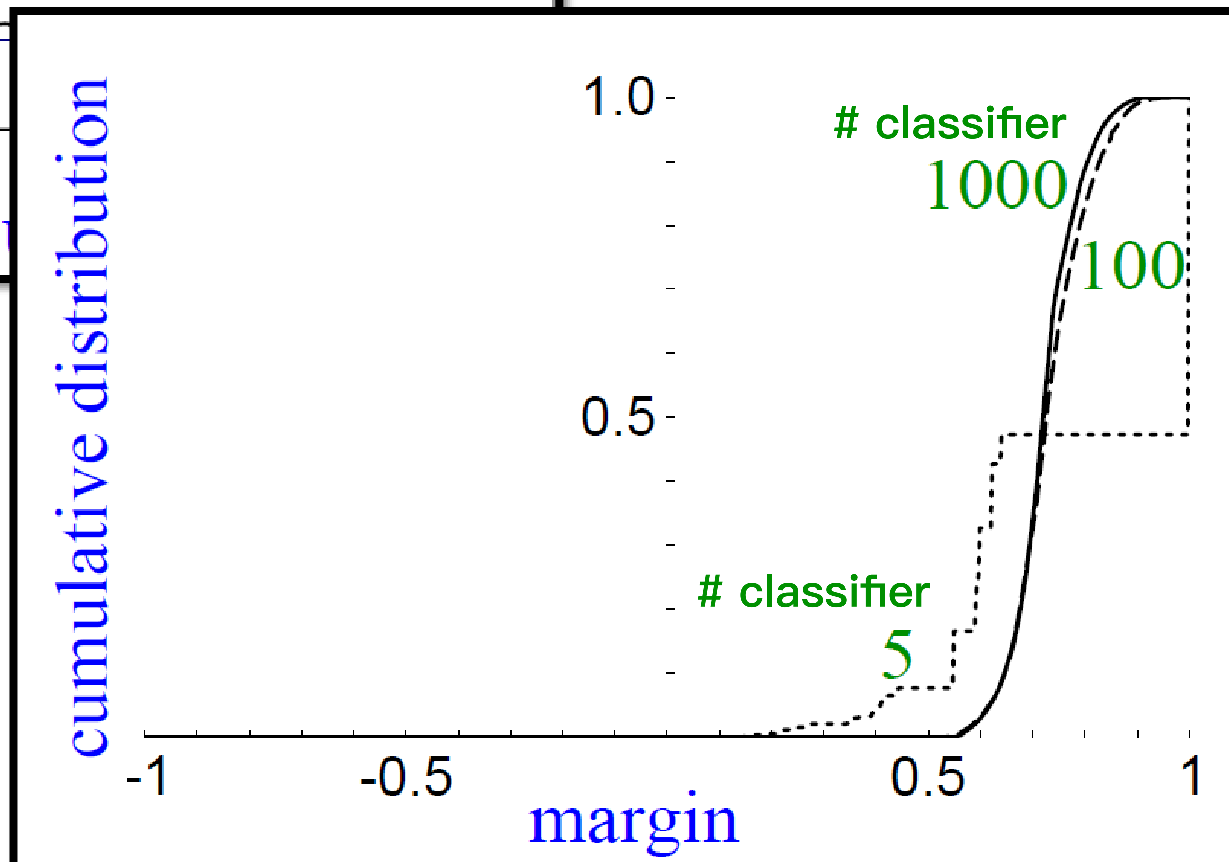


combined model

$$H(x) = sign\left(\underbrace{\sum_{t=1}^{T} \alpha_t f_t(x)}_{g(x)}\right)$$

Margin = $\hat{y} g(x)$

與0的距離



為什麼可以增加margin？

Large Margin?

$$H(x) = sign\left(\underbrace{\sum_{t=1}^{T} \alpha_t f_t(x)}_{g(x)}\right)$$

Training Data Error Rate =

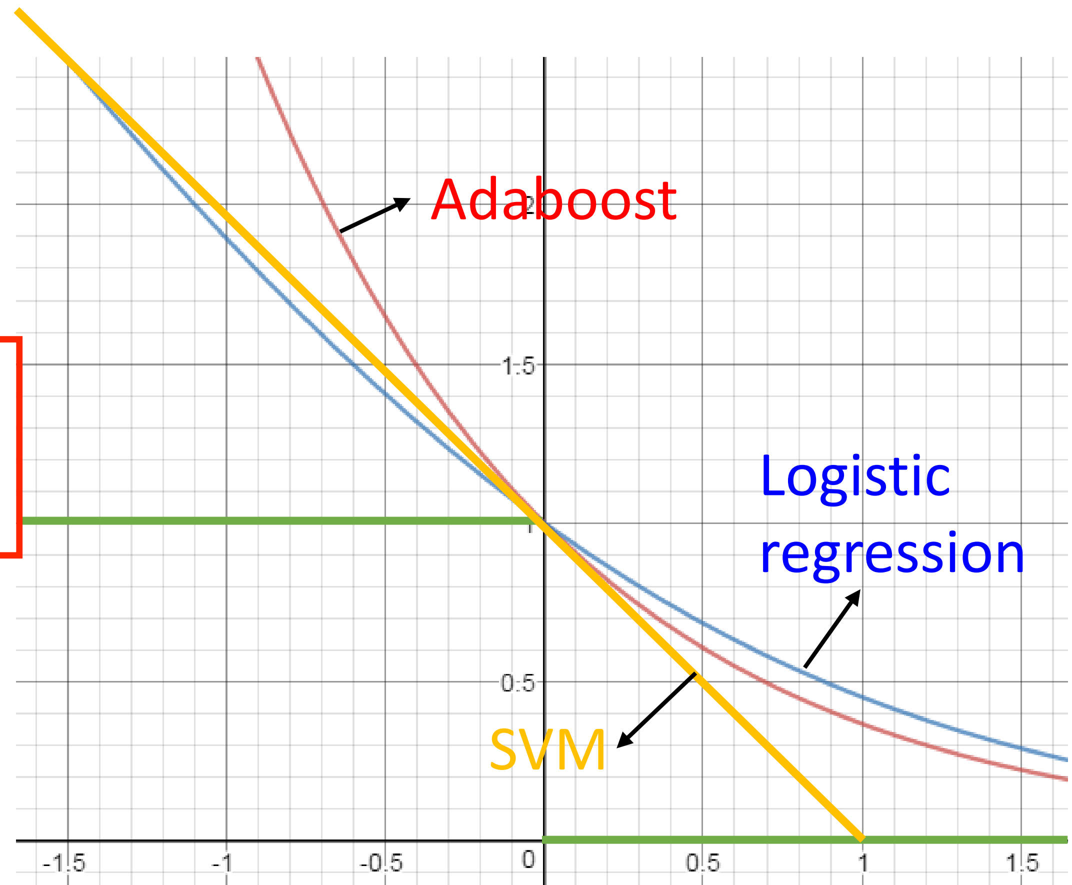
$$= \frac{1}{N}\sum_{n} \delta(H(x^n) \neq \hat{y}^n)$$

$$\leq \boxed{\frac{1}{N}\sum_{n} exp(-\hat{y}^n g(x^n))}$$

adaboost

$$= \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)}$$

Getting smaller and smaller as T increase



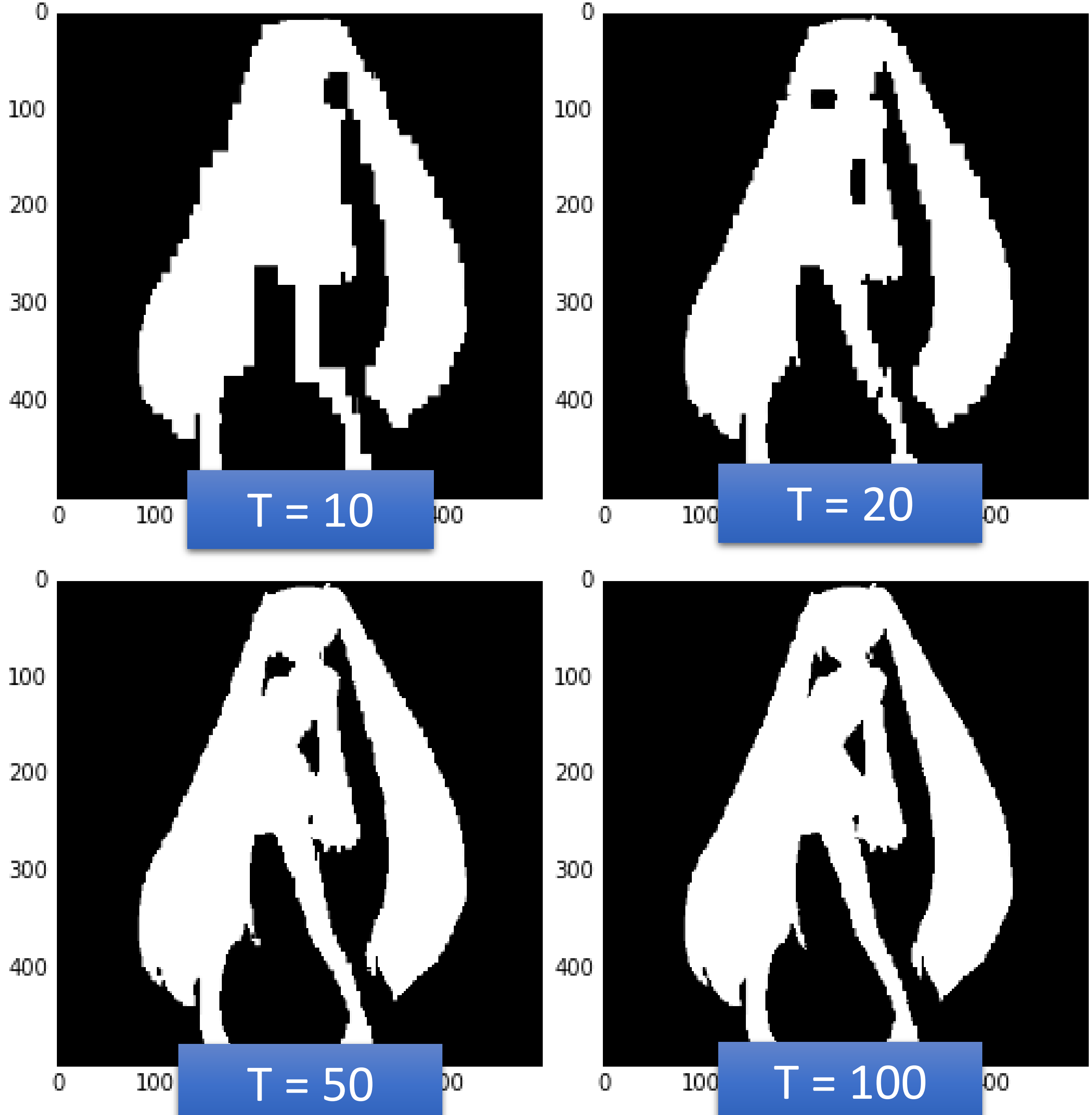
$\hat{y}^n g(x^n)$

就算error rate=0，對於adaboost在往右的同時error可以在更小

Experiment: Function of Miku

Adaboost
+Decision Tree

(depth = 5)



To learn more …

- Introduction of Adaboost:
  - Freund; Schapire (1999). "A Short Introduction to Boosting“
- Multiclass/Regression
  - Y. Freund, R. Schapire, “A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting”, 1995.
  - Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pages 80–91, 1998.
- Gentle Boost
  - Schapire, Robert; Singer, Yoram (1999). "Improved Boosting Algorithms Using Confidence-rated Predictions".

General Formulation of Boosting

- Initial function $g_0(x) = 0$
- For t = 1 to T:
  - Find a function $f_t(x)$ and α_t to $\boxed{\text{improve}}$ $g_{t-1}(x)$ weight
    - $g_{t-1}(x) = \sum_{i=1}^{t-1} \alpha_i f_i(x)$
  - $g_t(x) = g_{t-1}(x) + \alpha_t f_t(x)$
- Output: $H(x) = sign\big(g_T(x)\big)$

What is the learning target of $g(x)$? 如何找到ft(x)

將loss function定義如下

Minimize $L(g) = \sum_n \underbrace{l\big(\hat{y}^n, g(x^n)\big)}_{\text{loss function}} = \sum_n exp(-\hat{y}^n g(x^n))$

Gradient Boosting

- Find $\mathrm{g}(x)$, minimize $L(g) = \sum_n exp(-\hat{y}^n g(x^n)\,)$
  - If we already have $\mathrm{g}(x) = g_{t-1}(x)$, how to update $\mathrm{g}(x)$?

想像：改變g在某一個點的值對 L 的影響有多大

把function G對目標函式微分

Gradient Descent:

$$g_t(x) = g_{t-1}(x) \boxed{- \eta \left. \frac{\partial L(g)}{\partial g(x)} \right|_{\mathrm{g}(x) = g_{t-1}(x)}}$$

$$-\sum_n exp(-\hat{y}^n g_{t-1}(x^n)\,)(-\hat{y}^n)$$

Same direction

Boosting角度

$$g_t(x) = g_{t-1}(x) \boxed{+ \alpha_t f_t(x)}$$

Gradient Boosting

$$\overset{\text{vector}}{f_t(x)} \xleftrightarrow{\text{Same direction}} \overset{\text{vector}}{\sum_n exp(-\hat{y}^n g_t(x^n))(\hat{y}^n)}$$

We want to find $f_t(x)$ maximizing

判斷正確

Minimize Error

值越大代表方向越一致

$$\sum_n \underbrace{exp(-\hat{y}^n g_{t-1}(x^n))}_{\text{example weight } u_t^n} \underbrace{\boxed{(\hat{y}^n) f_t(x^n)}}_{\text{Same sign}}$$

$$u_t^n = exp(-\hat{y}^n g_{t-1}(x^n)) = exp\left(-\hat{y}^n \sum_{i=1}^{t-1} \alpha_i f_i(x^n)\right)$$

$$= \prod_{i=1}^{t-1} exp(-\hat{y}^n \alpha_i f_i(x^n))$$

Exactly the weights we obtain in Adaboost

Gradient Boosting

先找出ft後固定住，窮舉所有alpha（learning rate）讓loss最小

- Find g(x), minimize $L(g) = \sum_n exp(-\hat{y}^n g(x^n))$

$$g_t(x) = g_{t-1}(x) + \alpha_t f_t(x)$$

α_t is something like learning rate

Find α_t minimzing $L(g_{t+1})$

$$L(g) = \sum_n exp(-\hat{y}^n(g_{t-1}(x) + \alpha_t f_t(x)))$$

$$= \sum_n exp(-\hat{y}^n g_{t-1}(x)) exp(-\hat{y}^n \alpha_t f_t(x))$$

$$= \sum_{\hat{y}^n \neq f_t(x)} exp(-\hat{y}^n g_{t-1}(x^n)) exp(\alpha_t)$$

$$+ \sum_{\hat{y}^n = f_t(x)} exp(-\hat{y}^n g_{t-1}(x^n)) exp(-\alpha_t)$$

Find α_t such that

$$\frac{\partial L(g)}{\partial \alpha_t} = 0$$

$$\alpha_t = ln\sqrt{(1-\varepsilon_t)/\varepsilon_t}$$

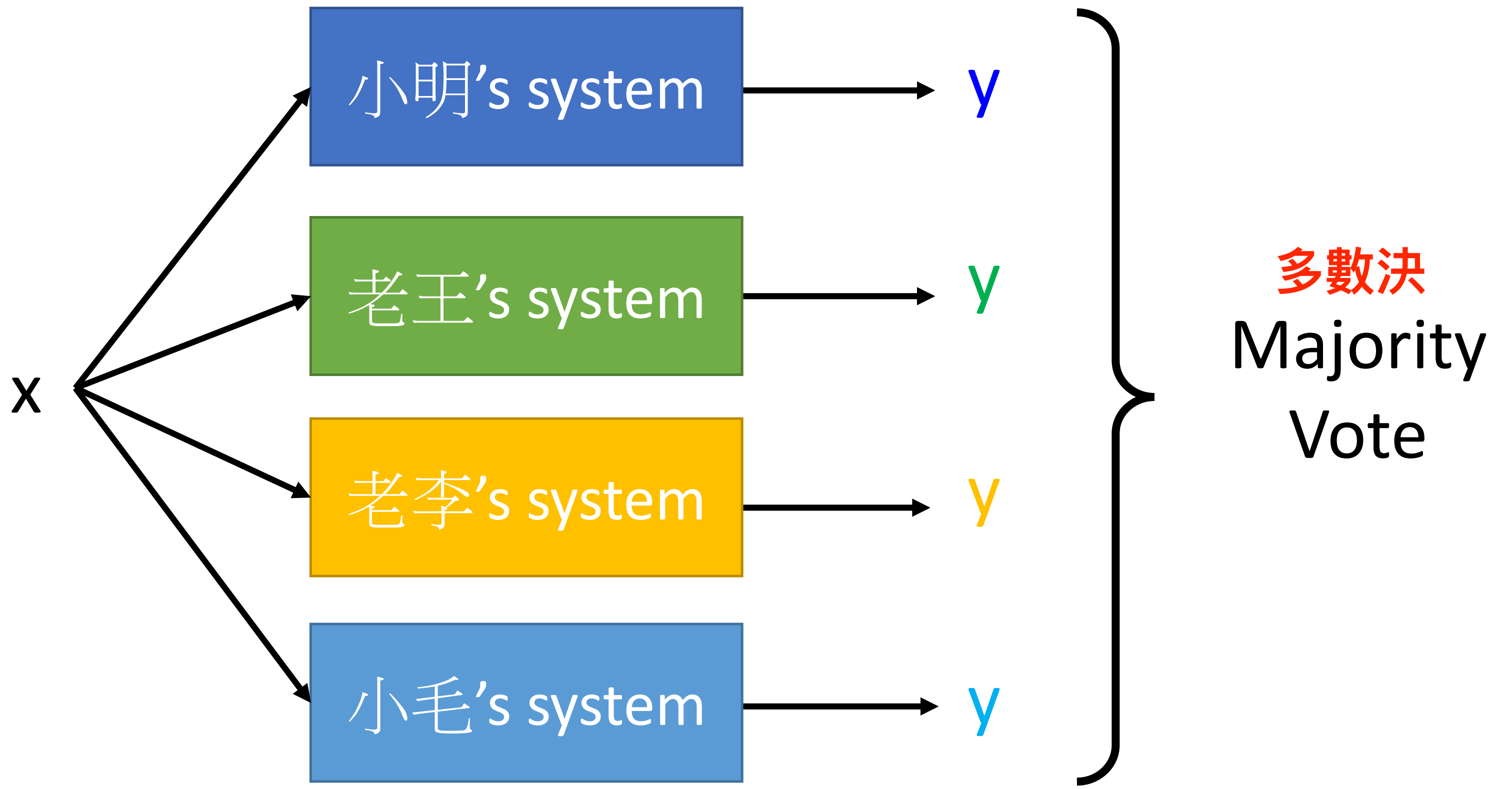
Adaboost!

Cool Demo

- http://arogozhnikov.github.io/2016/07/05/gradient_boosting_playground.html

Ensemble: Stacking

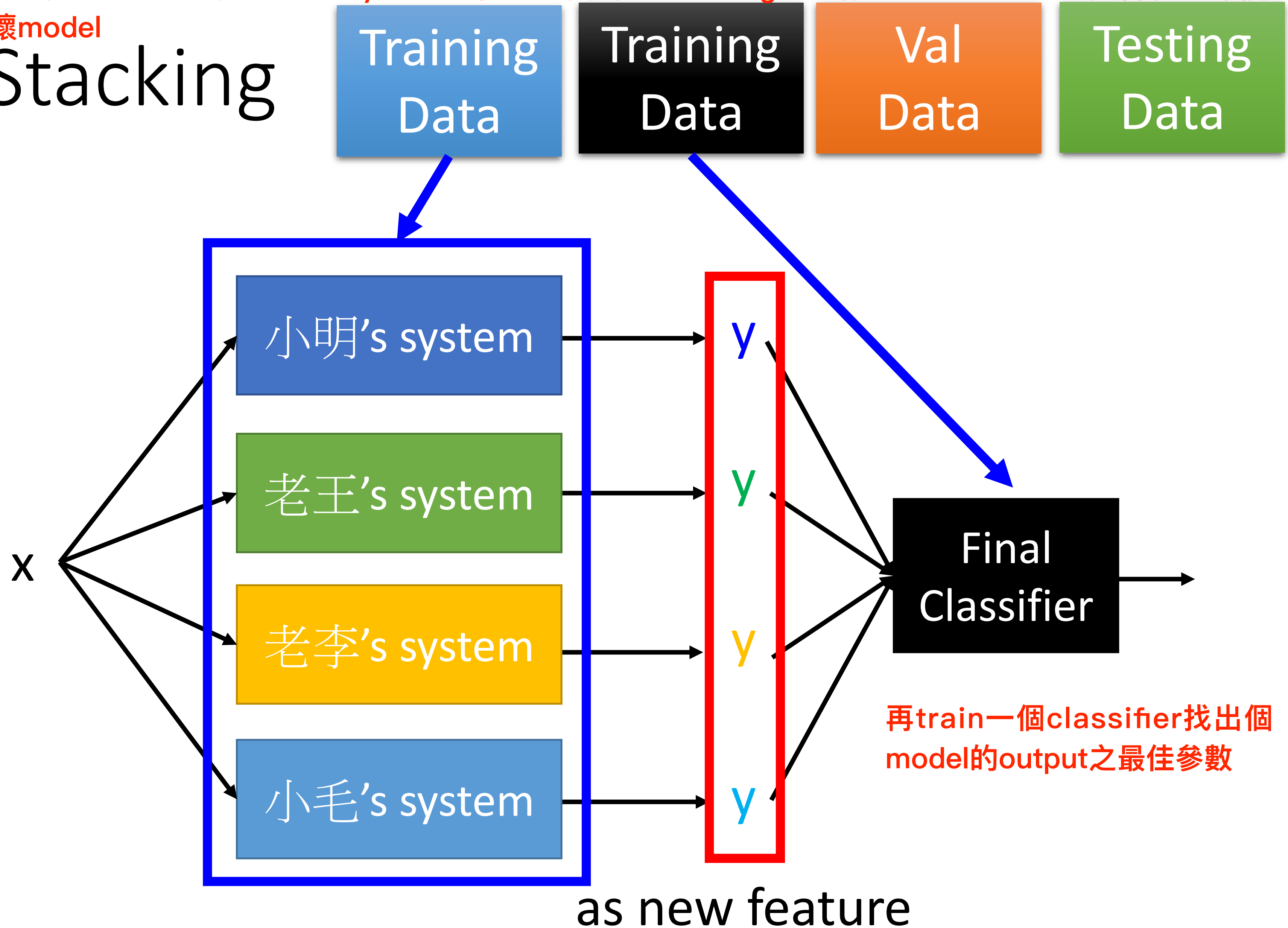
Voting



x
小明's system
老王's system
老李's system
小毛's system
y
y
y
y
多數決
Majority Vote

然而這樣做需要將training data分兩群，一部份train各自的model，一部分拿來做final classifier的training，避免有些異常的model雖然accuracy=100%但是是因為overfitting，這樣final classifier只會參考那個fit 100%的壞model

Stacking



再train一個classifier找出個model的output之最佳參數