Ensemble Learning

Lecture 7, DD2431 Machine Learning

Josephine Sullivan.

September 26, 2011

Today's lecture: Ensemble Learning

We will describe and investigate algorithms to

train weak classifiers/regressors and how to combine them

to construct a classifier/regressor more powerful than any of the individual ones.

Today's lecture: Ensemble Learning

Outline of the lecture:

Motivation Wisdom of Crowds

Classifier characterization

Why combine classifiers?

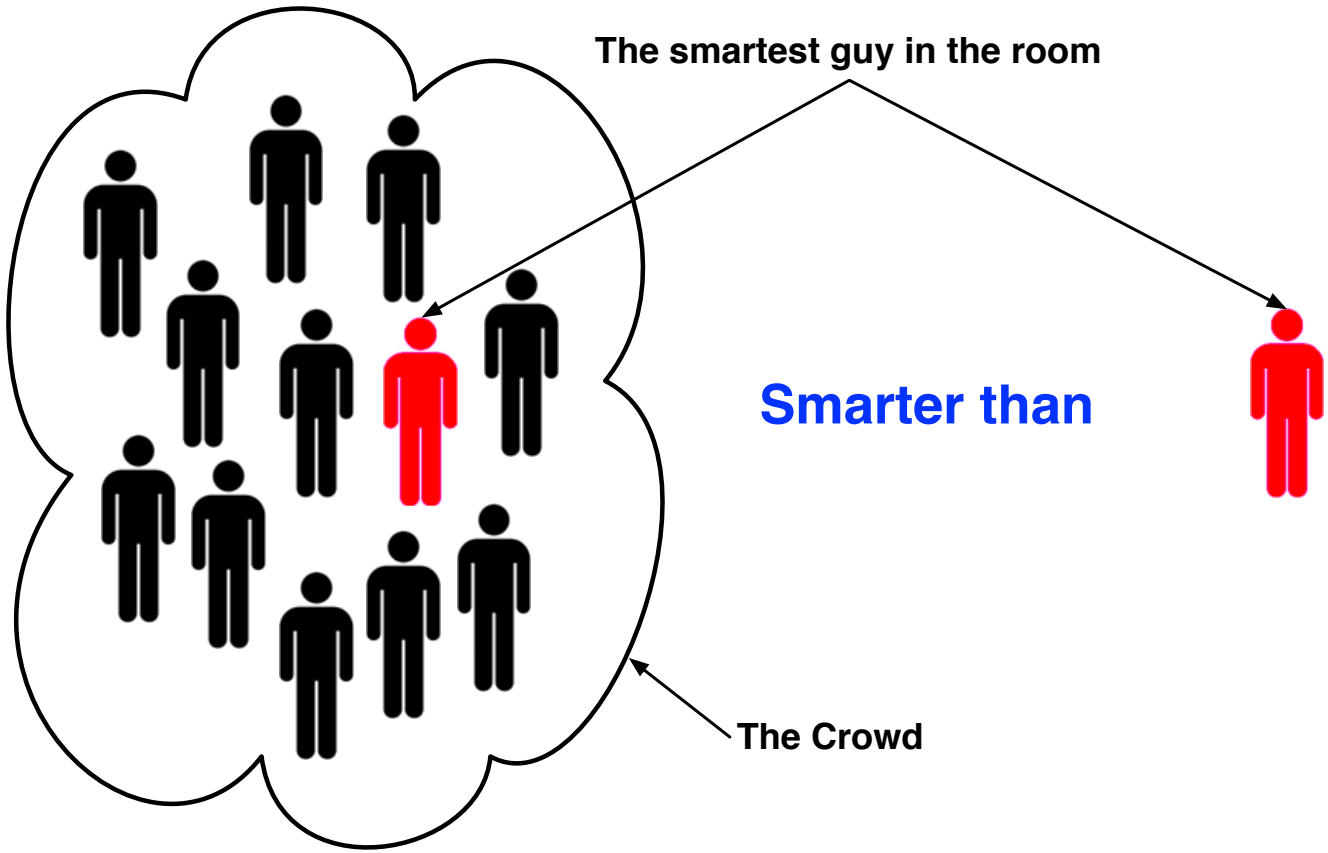
Ensemble I - Boosting Listen this is the basis for next lab!

Ensemble II - Bagging (for reference only)

Hierarchical Mixture of Experts (for reference only)

Final Summary

The Wisdom of Crowds



The **collective knowledge** of a *diverse* and *independent* body of people typically **exceeds** the knowledge of **any single individual** and can be harnessed by voting.

The Wisdom of Crowds - Really?

Crowd wiser than **any individual**

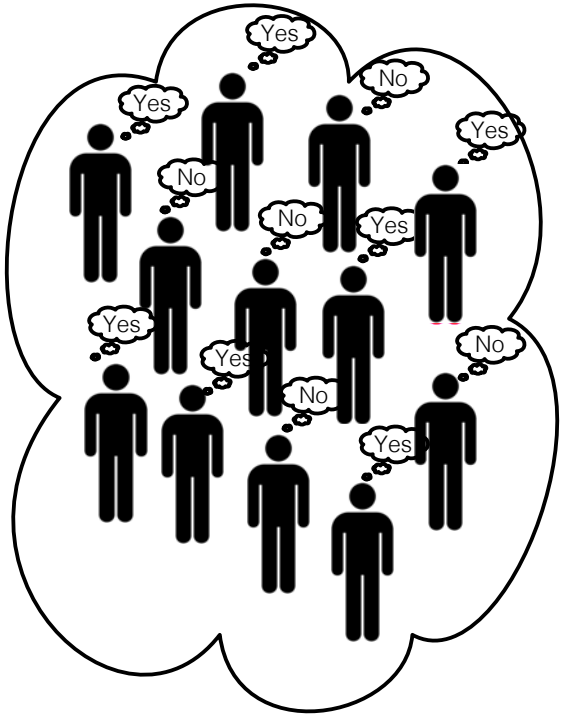
- When ?
- For which questions ?

See **The Wisdom of Crowds** by *James Surowiecki* published in 2004 to see this idea applied to business.

Consider this scenario

Ask each person in the crowd:

Will Mr. X win the general election in country Y?



Crowd's prediction:

MAJORITY answer.

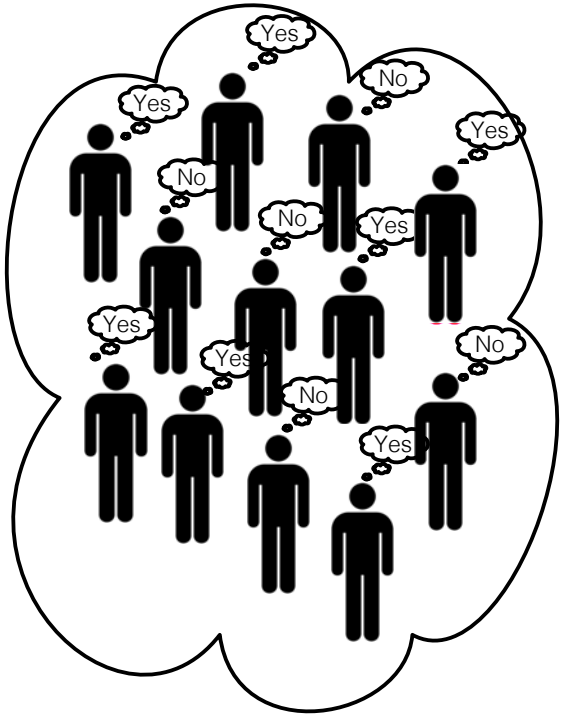
⇐ This crowd predicts Yes.

(Mr. X will win the election.)

Consider this scenario

Ask each person in the crowd:

Will Mr. X win the general election in country Y?



Crowd's prediction:

MAJORITY answer.

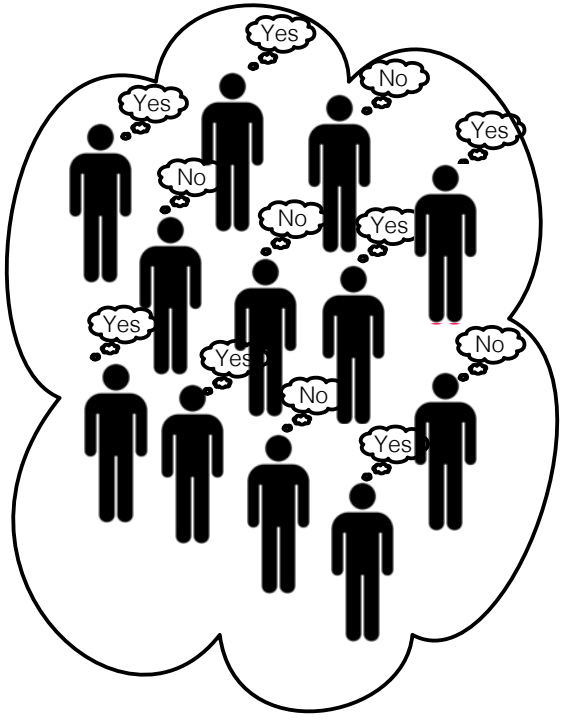
$\Leftarrow$ This crowd predicts **Yes**.

(Mr. X will win the election.)

Consider this scenario

Ask each person in the crowd:

Will Mr. X win the general election in country Y?



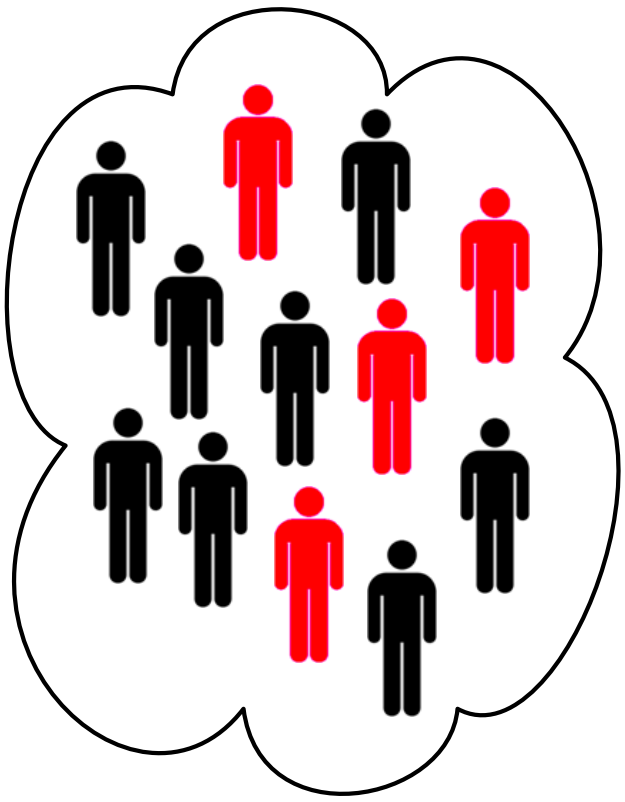
Crowd's prediction:

MAJORITY answer.

$\Leftarrow$ This crowd predicts **Yes**.

(Mr. X will win the election.)

Has crowd made a good prediction?



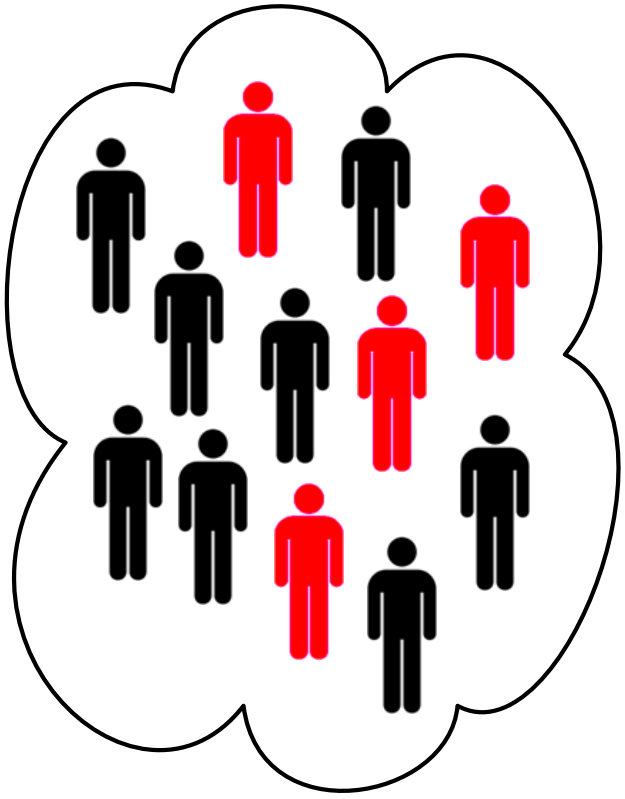
If composition of crowd:

30% **EXPERTS**.
70% **NON-EXPERTS**.

and their level of expertise:

$$P(\text{correct prediction} \,|\, \text{expert}) = p_e$$
$$P(\text{correct prediction} \,|\, \text{non-expert}) = p_{ne}$$

Has crowd made a good prediction?



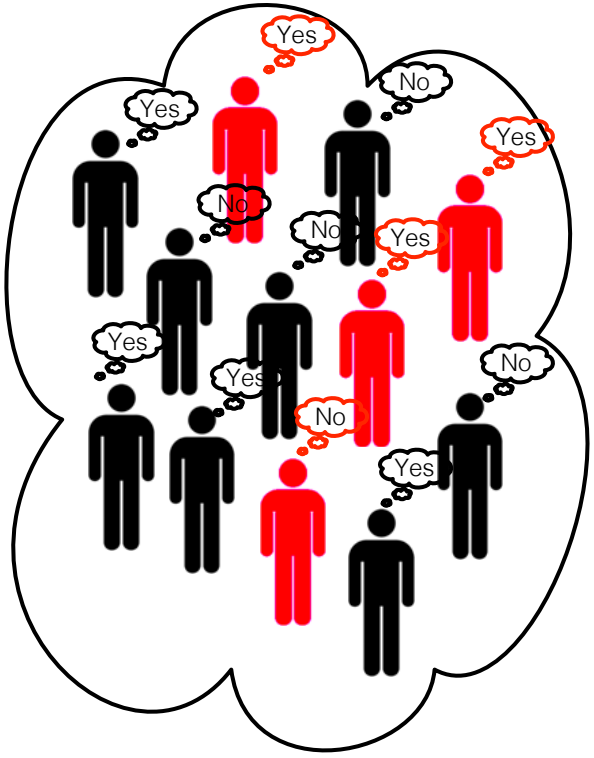
If composition of crowd:

30% **EXPERTS**.
70% **NON-EXPERTS**.

and their level of expertise:

$$P(\text{correct prediction} \,|\, \text{expert}) = p_e$$
$$P(\text{correct prediction} \,|\, \text{non-expert}) = p_{ne}$$

Has crowd made a good prediction?

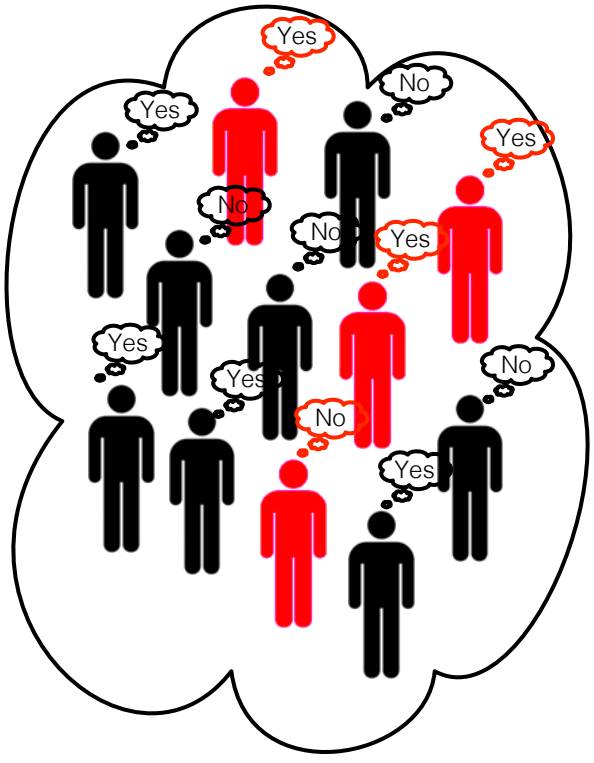


Let $p_e = .8$ and $p_{ne} = .5$

For random person from crowd:

$$P(\text{correct pred.} \mid \text{individual}) = .3\,p_e + .7\,p_{ne}$$
$$= .59$$

Has crowd made a good prediction?



Let $p_e = .8$ and $p_{ne} = .5$

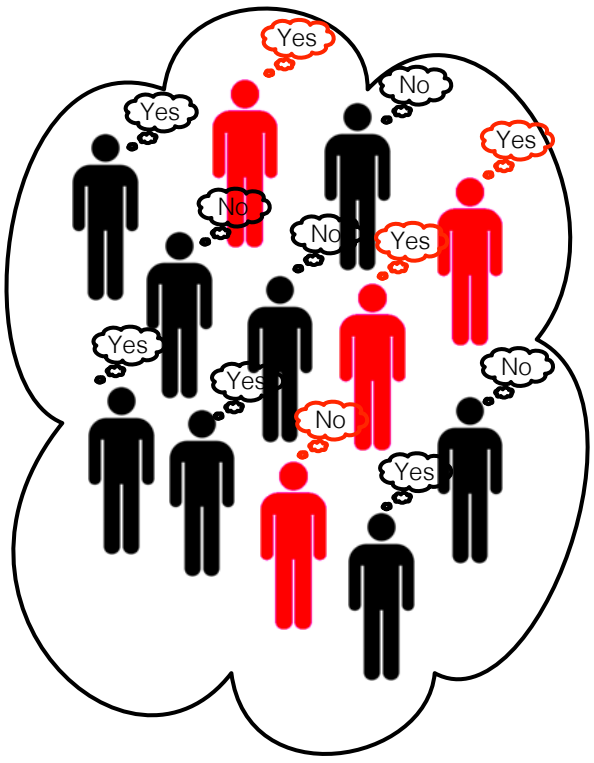
$P(\text{correct pred.}|\text{individual}) = p_i = .59$

If crowd contains 50 *independent* **people:**

$$P(\text{correct pred.}|\text{crowd}) = \sum_{k=26}^{50} \binom{50}{k} p_i^k (1-p_i)^{50-k}$$

$$= .8745$$

Has crowd made a good prediction?



Let $p_e = .8$ and $p_{ne} = .5$

$P(\text{correct pred.}|\text{individual}) = p_i = .59$

If crowd contains 50 *independent* **people:**

$$P(\text{correct pred.}|\text{crowd}) = \sum_{k=26}^{50} \binom{50}{k} p_i^k (1-p_i)^{50-k}$$

$$= .8745$$

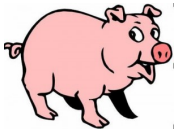
This crowd has made a prediction with probability .875 of being correct which is $> p_e$.

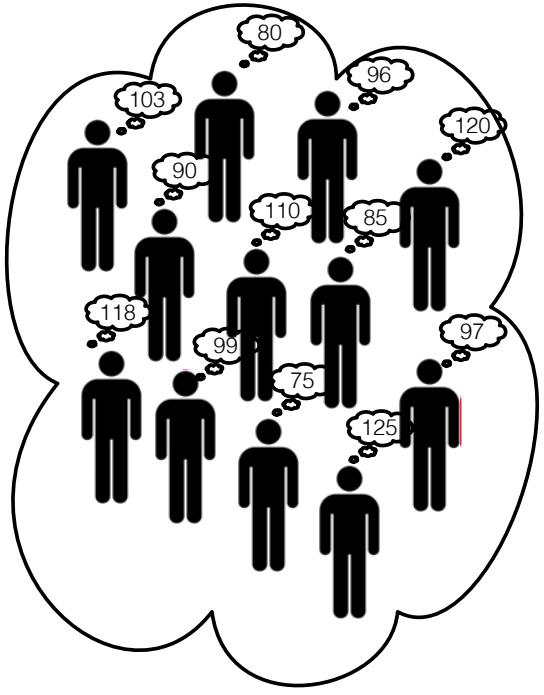
It is wiser than each of the experts!

Another scenario

Ask each person in the same crowd:

How much does the pig weigh?





Crowd's prediction:

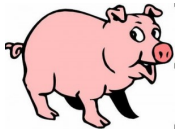
AVERAGE of all predictions.

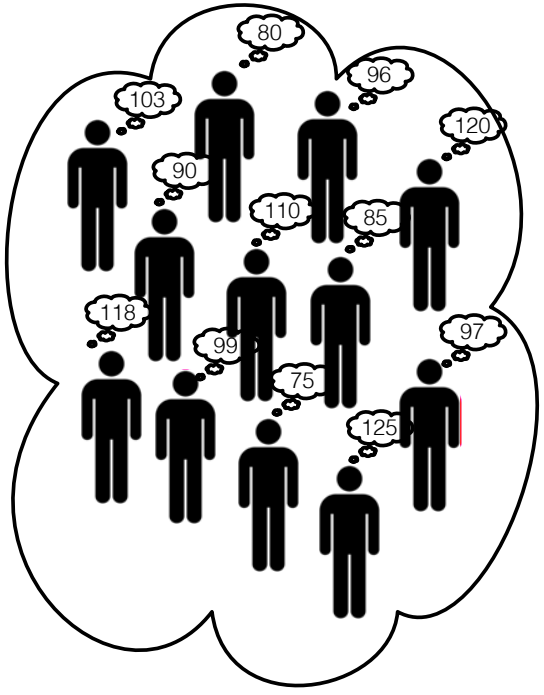
← This crowd predicts **99.8333**.

(The pig weighs 99.8333 kg.)

Another scenario

Ask each person in the same crowd:

How much does the pig weigh? 



Crowd's prediction:

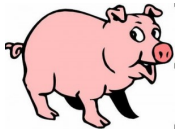
AVERAGE of all predictions.

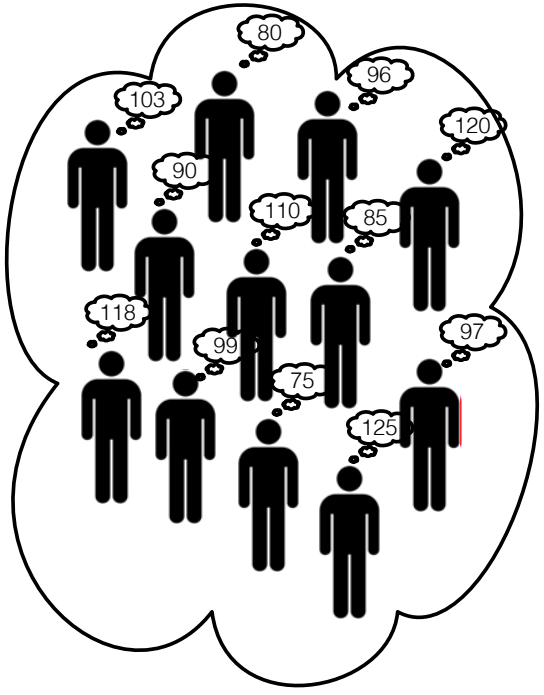
$\Leftarrow$ This crowd predicts **99.8333**.

(The pig weighs 99.8333 kg.)

Another scenario

Ask each person in the same crowd:

How much does the pig weigh? 



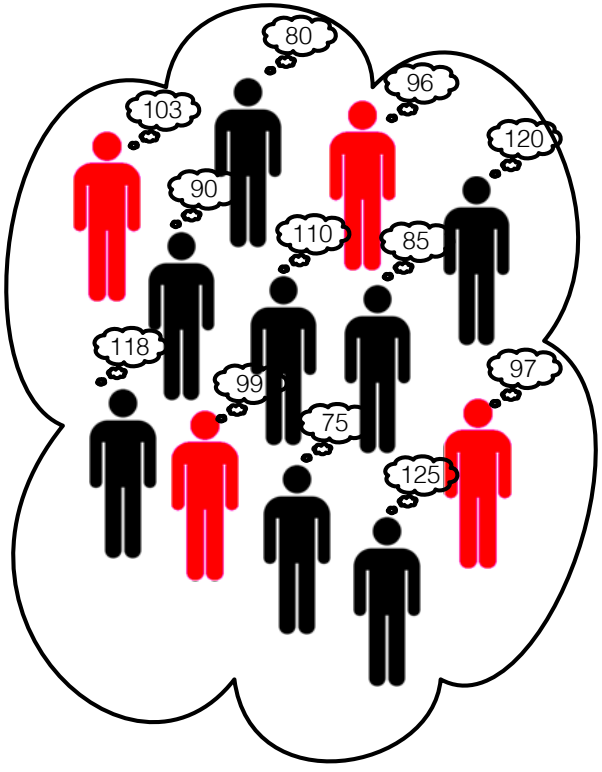
Crowd's prediction:

AVERAGE of all predictions.

$\Leftarrow$ This crowd predicts **99.8333**.

(The pig weighs 99.8333 kg.)

Has crowd made a good estimate?

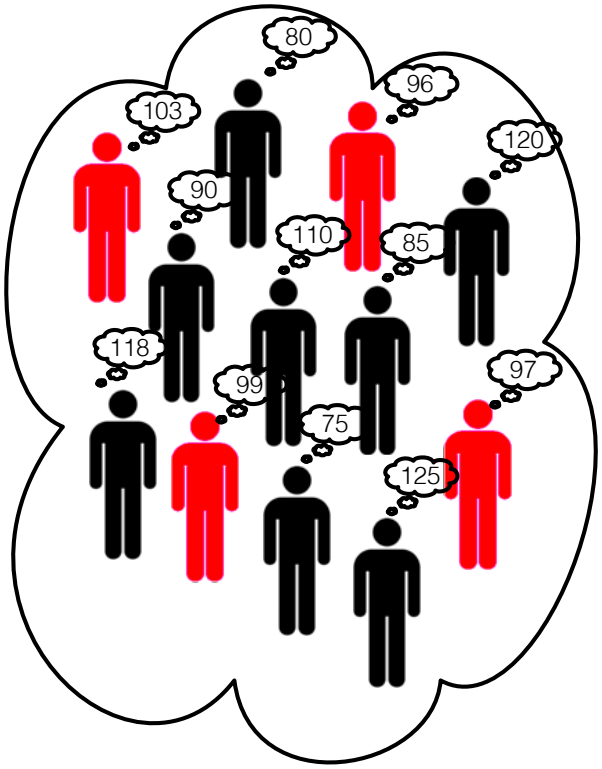


If composition of crowd:

30% **EXPERTS**.

70% **NON-EXPERTS**.

Has crowd made a good estimate?



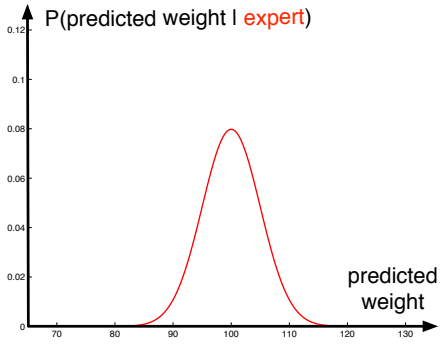
If composition of crowd:

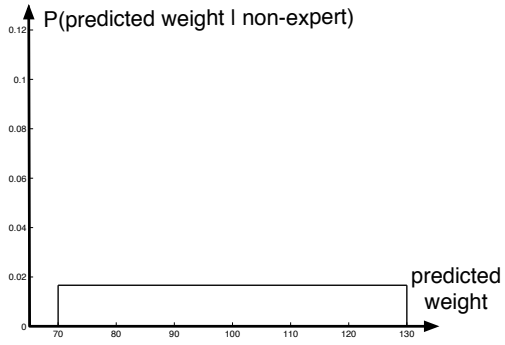
30% EXPERTS.

70% NON-EXPERTS.

and their level of expertise:

(Say *pig*'s true weight is 100 kg)





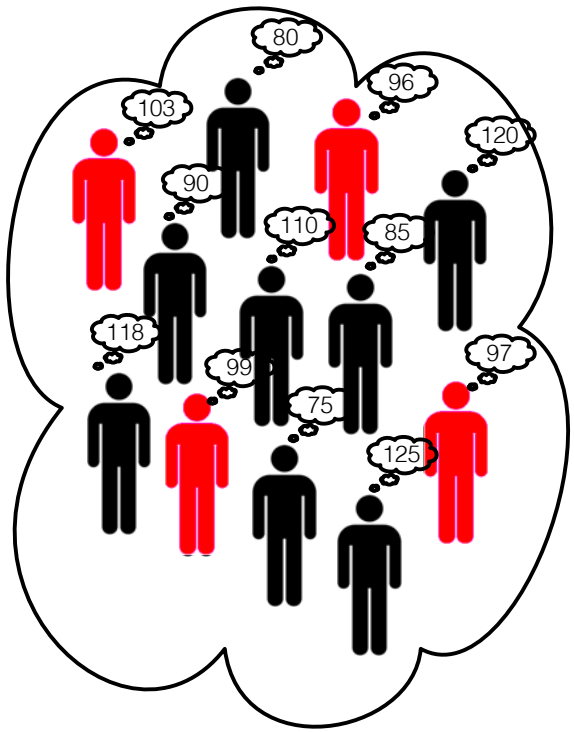
$P(\text{pred. weight} \mid \text{expert})$:

$\mathcal{N}(100, 5^2)$

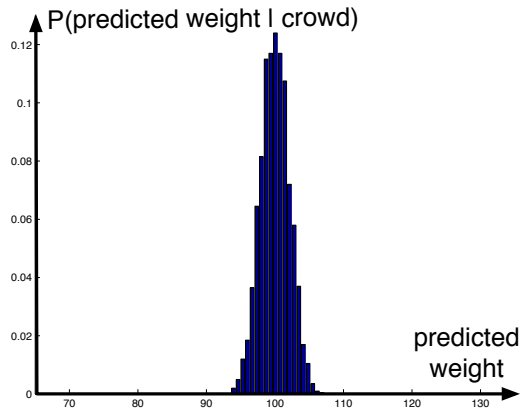
$P(\text{pred. weight} \mid \text{non-expert})$:

$\mathcal{U}(70, 130)$

Has crowd made a good estimate?



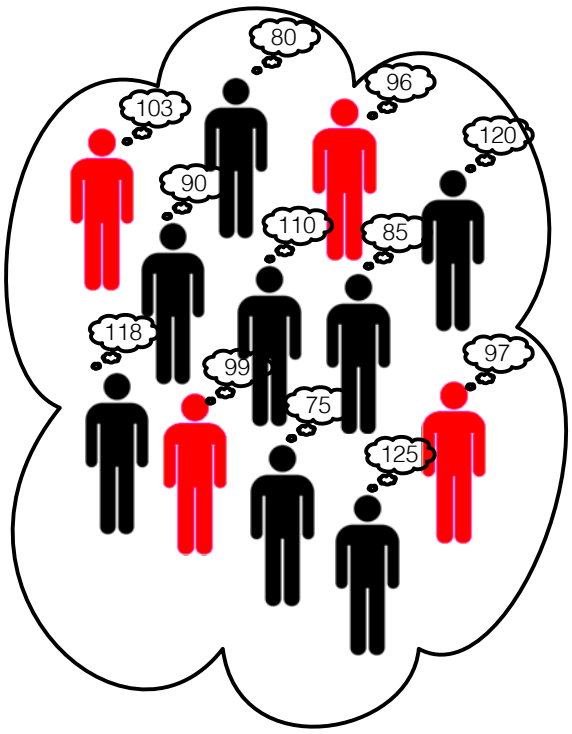
If crowd contains *independent* **50 people:**



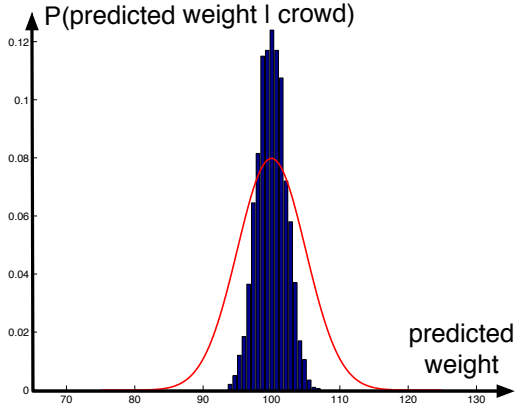
$\Uparrow$

$P(\text{pred. weight}|\text{crowd})$

Has crowd made a good estimate?



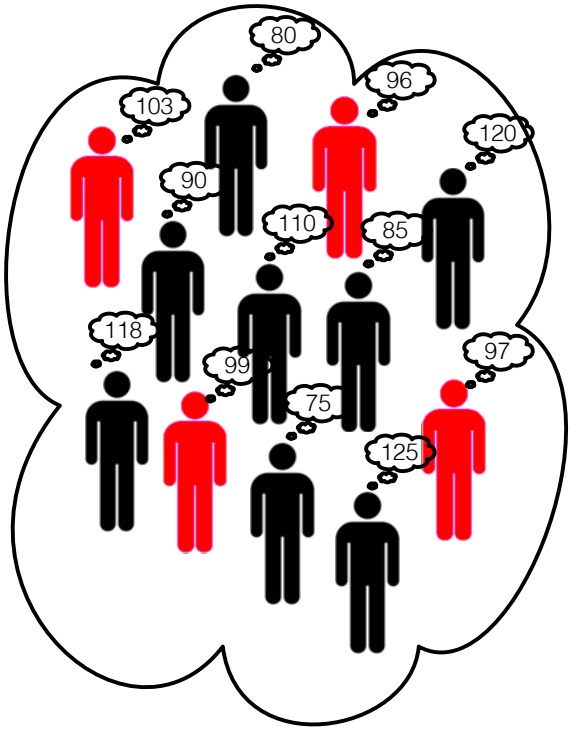
If crowd contains *independent* **50 people:**



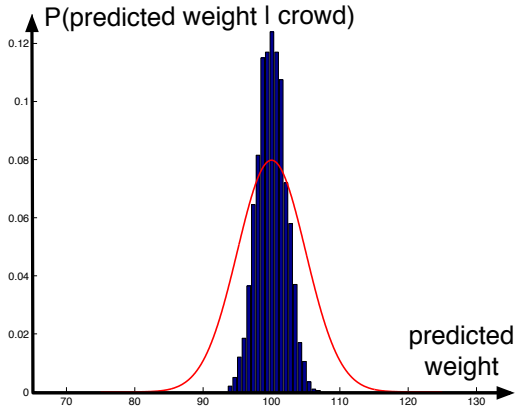
$\Uparrow$

$P(\text{pred. weight}|\text{crowd})$ **and**
$P(\text{pred. weight}|\text{expert})$

Has crowd made a good estimate?



If crowd contains *independent* **50 people:**



$\Uparrow$

$P(\text{pred. weight}|\text{crowd})$ **and**
$P(\text{pred. weight}|\text{expert})$

On average this crowd will make better estimates than the experts.

It is wiser than each of the experts!

But....

Why didn't I just asked a bunch of experts??

- Large enough crowd $\Longrightarrow$ high probability a sufficient number of experts will be in crowd (for any question).
- Random selection $\Longrightarrow$ don't make a biased choice in experts.
- For some questions it may be hard to identify a diverse set of experts

But....

Why didn't I just asked a bunch of experts??

- Large enough crowd $\implies$ high probability a sufficient number of experts will be in crowd (for any question).
- Random selection $\implies$ don't make a biased choice in experts.
- For some questions it may be hard to identify a diverse set of experts

But....

Why didn't I just asked a bunch of experts??

- Large enough crowd $\Longrightarrow$ high probability a sufficient number of experts will be in crowd (for any question).
- Random selection $\Longrightarrow$ don't make a biased choice in experts.
- For some questions it may be hard to identify a diverse set of experts

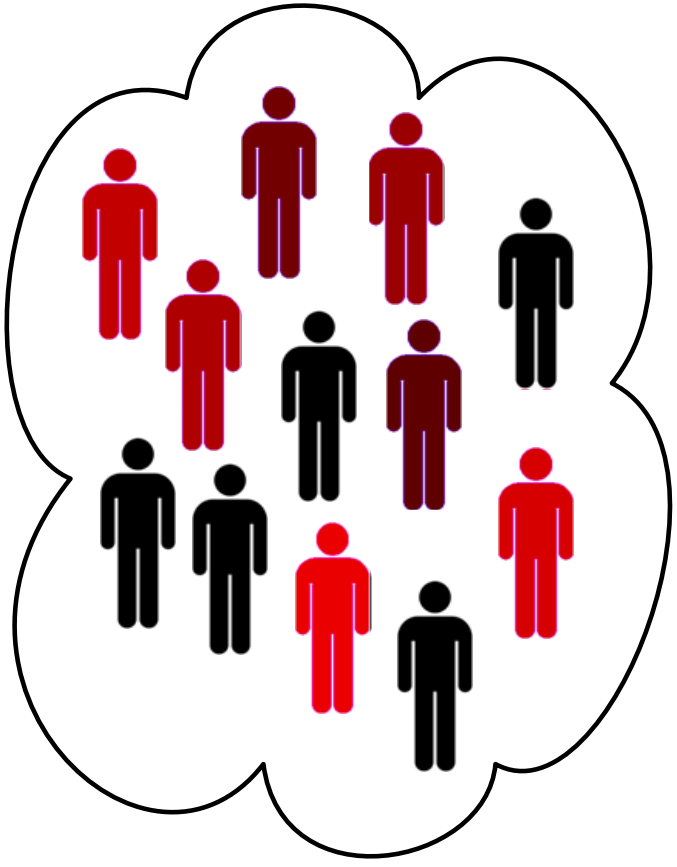
But....

Why didn't I just asked a bunch of experts??

- Large enough crowd $\Longrightarrow$ high probability a sufficient number of experts will be in crowd (for any question).
- Random selection $\Longrightarrow$ don't make a biased choice in experts.
- For some questions it may be hard to identify a diverse set of experts

For a random crowd

Given a **random question** expect each **person** to have a **different level** of **expertise**.

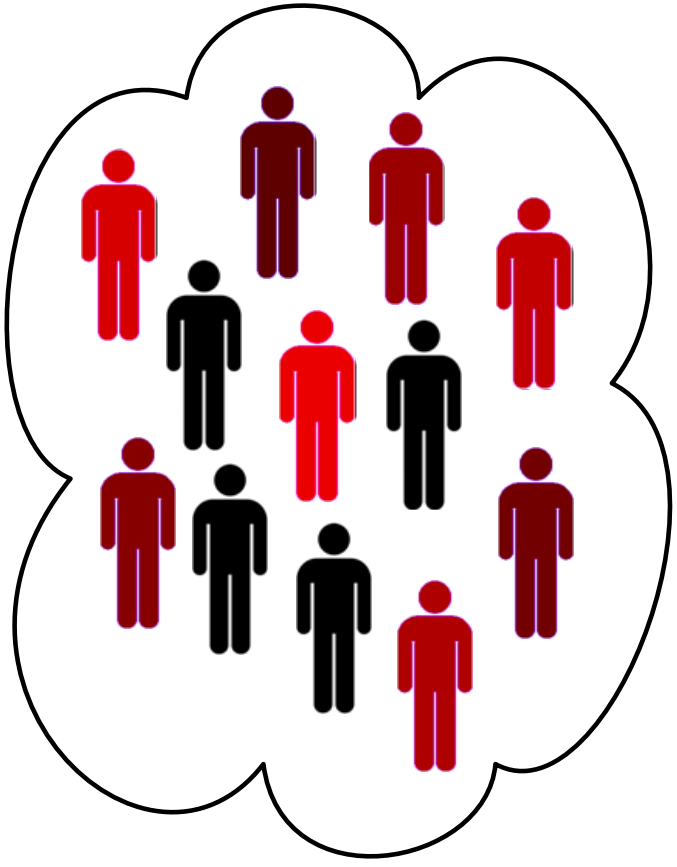


Will it rain tomorrow?

$\Leftarrow$ redness proportional to expertise.

For a random crowd

Given a **random question** expect each **person** to have a **different level** of **expertise**.



Will Obama be re-elected as president?

$\Leftarrow$ redness proportional to expertise.

What makes a crowd wise?

According to *James Surowiecki* there are four elements required to form a wise crowd

- **Diversity of opinion.** People in crowd should have a range of experiences, education and opinions.
  (Encourages independent predictions)
- **Independence.** Prediction by person in crowd is not influenced by other people in the crowd.
- **Decentralization** People have specializations and local knowledge.
- **Aggregation.** There is a mechanism for aggregating all predictions into one single prediction.

What makes a crowd wise?

According to *James Surowiecki* there are four elements required to form a wise crowd

- **Diversity of opinion.** People in crowd should have a range of experiences, education and opinions.
  (Encourages independent predictions)
- **Independence.** Prediction by person in crowd is not influenced by other people in the crowd.
- **Decentralization** People have specializations and local knowledge.
- **Aggregation.** There is a mechanism for aggregating all predictions into one single prediction.

What makes a crowd wise?

According to *James Surowiecki* there are four elements required to form a wise crowd

- **Diversity of opinion.** People in crowd should have a range of experiences, education and opinions.
  (Encourages independent predictions)
- **Independence.** Prediction by person in crowd is not influenced by other people in the crowd.
- **Decentralization** People have specializations and local knowledge.
- **Aggregation.** There is a mechanism for aggregating all predictions into one single prediction.

What makes a crowd wise?

According to *James Surowiecki* there are four elements required to form a wise crowd

- **Diversity of opinion.** People in crowd should have a range of experiences, education and opinions.
  (Encourages independent predictions)
- **Independence.** Prediction by person in crowd is not influenced by other people in the crowd.
- **Decentralization** People have specializations and local knowledge.
- **Aggregation.** There is a mechanism for aggregating all predictions into one single prediction.

The crowd must be careful

In the analysis of the crowd it is implicitly assumed:

- each person is not concerned with the opinions of others,
- no-one is copying anyone else in the crowd.

The crowd must be careful

In the analysis of the crowd it is implicitly assumed:

- each person is not concerned with the opinions of others,
- no-one is copying anyone else in the crowd.

If this is not adhered to the crowd runs the risk of...

Rational bubbles! Crowds can also be stupid

Rational bubbles have occurred when the crowd has had very bad judgment:

- *Tulip mania*, Netherlands 1630's,
- *Tech stock bubble*, 1990's,
- *Housing bubble*, Ireland in the 2000's,
- *Ponzi schemes*, Ivar Kreuger (famous KTH graduate), Bernie Madoff et al.

See **Extraordinary Popular Delusions and the Madness of Crowds** by Charles Mackay from the 1840's.

The crowd must be careful

In the analysis of the crowd I implicitly assumed:

- The non-experts will predict a completely random wrong answer - these will somewhat cancel each other out.
- However, there may be a systematic and consistent bias in the non-experts' predictions.

The crowd must be careful

In the analysis of the crowd I implicitly assumed:

- The non-experts will predict a completely random wrong answer - these will somewhat cancel each other out.
- However, there may be a systematic and consistent bias in the non-experts' predictions.
  This can lead to...

Wikiality! Crowds can also be stupid

If the crowd does not contain sufficient experts then *truth by consensus* (rather than fact) leads to **Wikiality**!

Term coined by *Stephen Colbert* in an episode of the *The Colbert Report* in July 2006.

Back to machines

Back to machine learning

This course considers different types of classifiers/regressors instead of a crowd of humans.

- None of these classifiers is as clever, flexible or has the wealth of experience as a human **but**
- their simplicity makes them easier to analyze !

Note: sometimes when I write *classifier* the idea also holds for *regressors*.

Back to machine learning

Will exploit *Wisdom of crowd* ideas for specific tasks by

- combining classifier predictions **and**
- aim to combine independent and diverse classifiers.

Back to machine learning

Will exploit *Wisdom of crowd* ideas for specific tasks by

- combining classifier predictions **and**
- aim to combine independent and diverse classifiers.

But will use labelled training data

- to identify the **expert** classifiers in the pool;
- to identify **complementary** classifiers;
- to indicate how to best combine them.

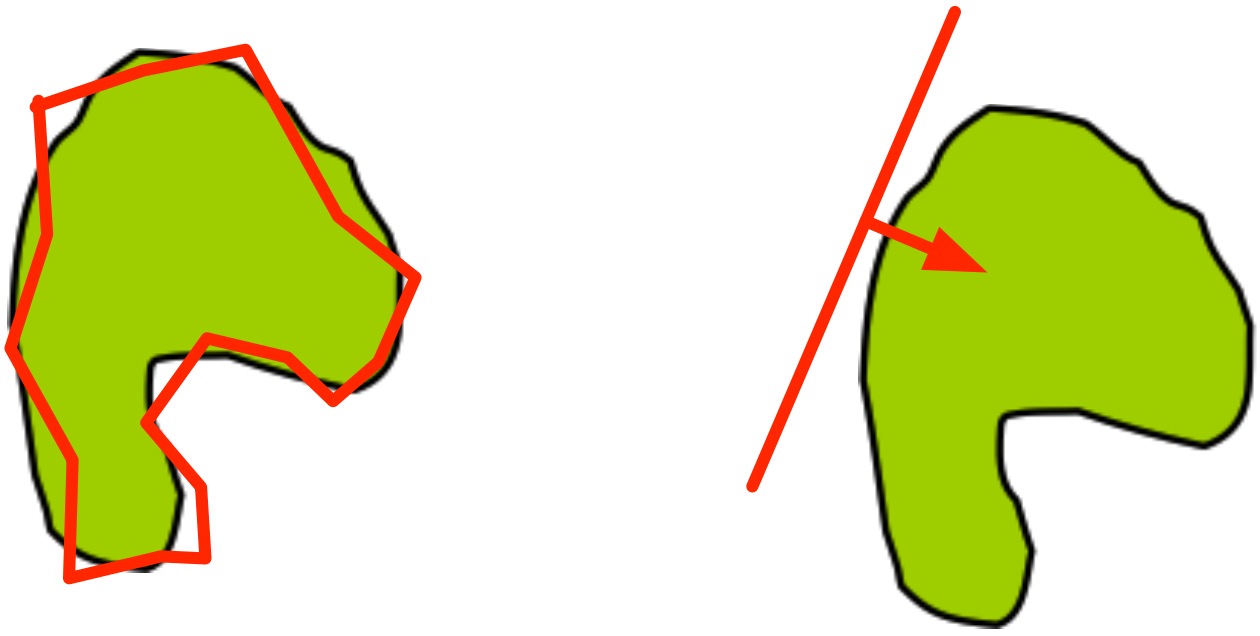
Characterization of a classifier: Bias

Bias of a classifier is the squared discrepancy between its averaged estimated and expected true function

$$(E[\hat{f}(\mathbf{x})] - E[f(\mathbf{x})])^2$$

Characterization of a classifier: Bias

Green region is the true boundary.



Low-bias classifier **High-bias classifier**

High model complexity (large # of d.o.f.) $\Longrightarrow$ Low-bias
Low model complexity (small # of d.o.f.) $\Longrightarrow$ High-bias

Ensemble Prediction: Voting

A **diverse** and **complementary** set of high-bias classifiers, with performance better than chance, combined by **voting**

$$f_V(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} h_t(\mathbf{x})\right)$$

can produce a classifier with a low-bias.

Example: Voting of oriented hyper-planes can define convex regions.

Ensemble Learning & Prediction

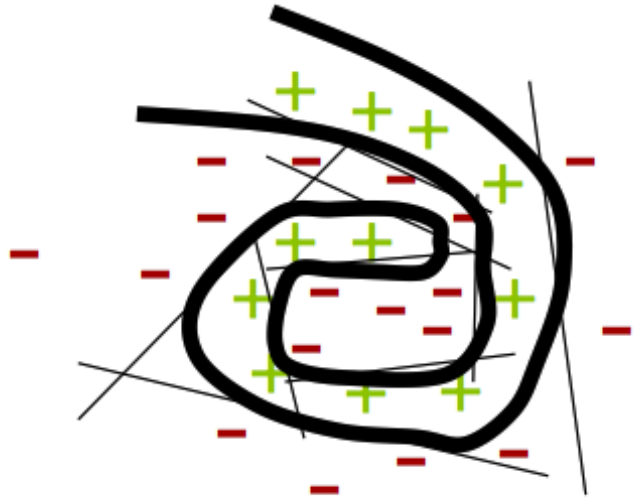
But how can we

- define a set of **diverse** and **complementary** high-bias classifiers, with non-random performance ?
- combine this set of high-biased classifiers to produce a low-bias classifier able to model a complex boundary (superior to voting)?

Ensemble Learning & Prediction

How? Exploit labelled training data.

- Train different classifiers using the training data which focus on different subsets of the data.
- Use a weighted sum of these *diversely* trained classifiers.



This approach allows simple high-bias classifiers to be combined to model very complex boundaries.

Algorithms: Boosting, Hierarchical Mixture of Experts

Characterization of a classifier: Variance

Variance of a classifier is the expected divergence of the estimated prediction function from its average value:

$$E[(\hat{f}(\mathbf{x}) - E[\hat{f}(\mathbf{x})])^2]$$

This measures how dependent the classifier is on the random sampling made in the training set.

Characterization of a classifier: Variance

Variance of a classifier is the expected divergence of the estimated prediction function from its average value:

$$E[(\hat{f}(\mathbf{x}) - E[\hat{f}(\mathbf{x})])^2]$$

This measures how dependent the classifier is on the random sampling made in the training set.

High model complexity (large # of d.o.f.) $\Longrightarrow$ High-variance
Low model complexity (small # of d.o.f.) $\Longrightarrow$ Low-variance

Characterization of a classifier: Variance

Variance of a classifier is the expected divergence of the estimated prediction function from its average value:

$$E[(\hat{f}(\mathbf{x}) - E[\hat{f}(\mathbf{x})])^2]$$

This measures how dependent the classifier is on the random sampling made in the training set.

Ensemble predictions such as bagging, voting, averaging using high-variance, low-bias classifiers reduce the variance of the ensemble classifier.

Ensemble method: **Boosting**

Ensemble Method: BOOSTING

Given: Training data $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ of inputs $\mathbf{x}_i$ and their labels $y_i \in \{-1, 1\}$ or real values.

$\mathcal{H}$ a family of possible weak classifiers/regression functions.

Output: A strong classifier/regression function

$$f_T(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right) \quad \text{or} \quad f_T(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

Ensemble Method: BOOSTING

Given: Training data $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ of inputs $\mathbf{x}_i$ and their labels $y_i \in \{-1, 1\}$ or real values.

$\mathcal{H}$ a family of possible weak classifiers/regression functions.

Output: A strong classifier/regression function

$$f_T(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right) \quad \text{or} \quad f_T(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

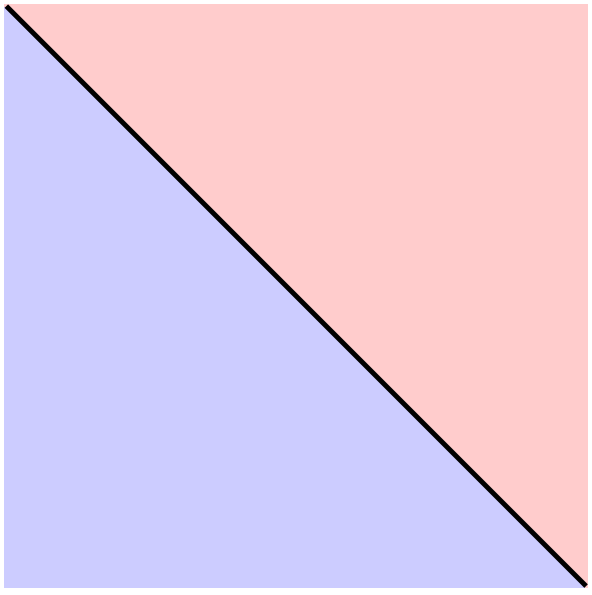
Ensemble Method: BOOSTING

How ?? (Just consider case of classification.)

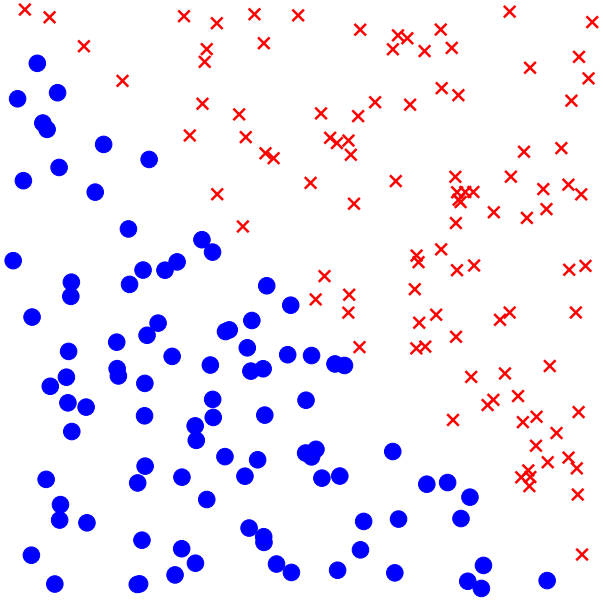
- Performance of classifiers $h_1, \ldots, h_t$ helps define h_{t+1}.
- Maintain weight $w_i^{(t)}$ for each training example in $\mathcal{S}$.
- Large $w_i^{(t)} \implies \mathbf{x}_i$ has greater influence on choice of h_t.
- Iteration t: $w_i^{(t)}$ increased if $\mathbf{x}_i$ wrongly classified by h_t.
- Iteration t: $w_i^{(t)}$ decreased if $\mathbf{x}_i$ correctly classified by h_t.

Remember: Each $h_t \in \mathcal{H}$

Binary classification example



True decision boundary

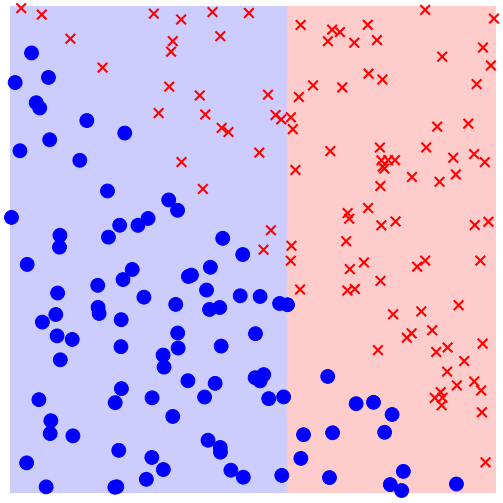


Training data

$\mathcal{H}$ is the set of all possible oriented vertical and horizontal lines.

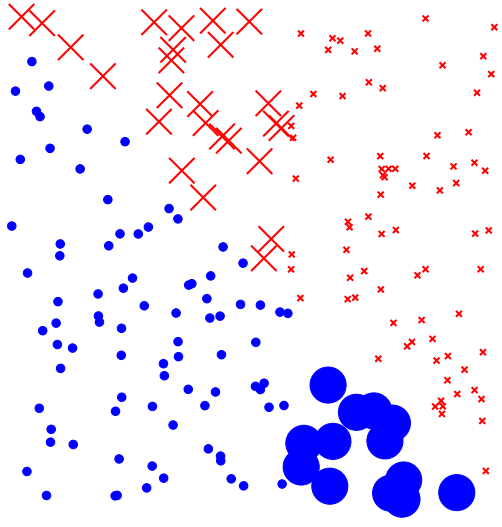
Example

Round 1



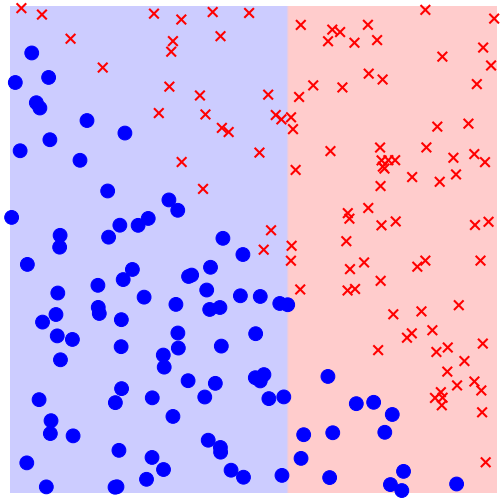
Chosen weak classifier

$\epsilon_1 = 0.19$, $\alpha_1 = 1.45$



Re-weight training points

$w_i^{(2)}$**'s**

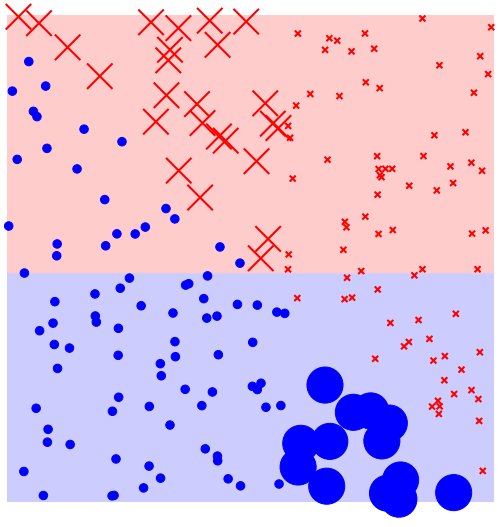


Current strong classifier

$f_2(\mathbf{x})$

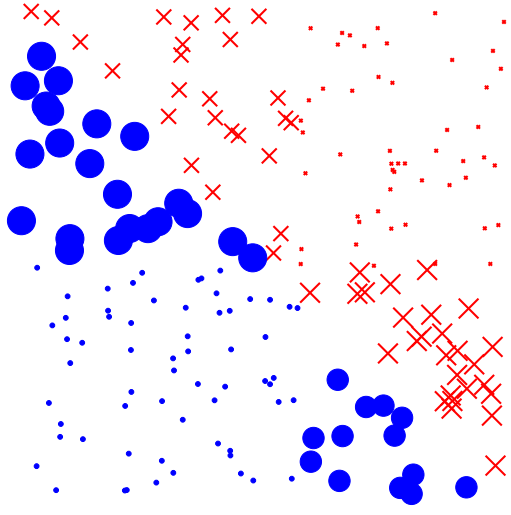
Example

Round 2



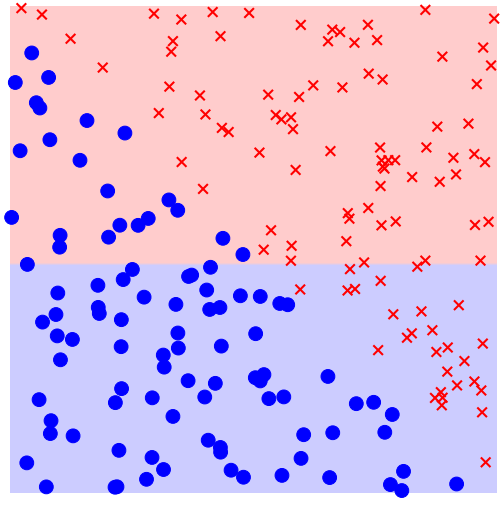
Chosen weak classifier

$\epsilon_2 = 0.1512,\ \alpha_2 = 1.725$



Re-weight training points

$w_i^{(3)}$**'s**

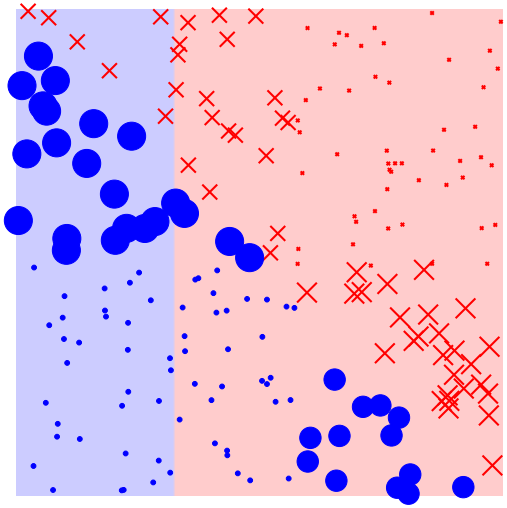


Current strong classifier

$f_2(\mathbf{x})$

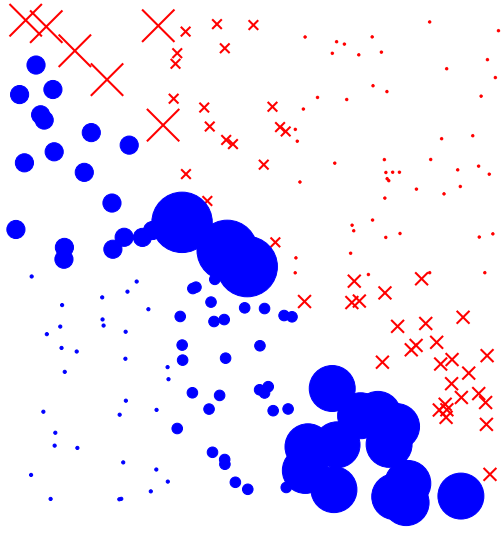
Example

Round 3



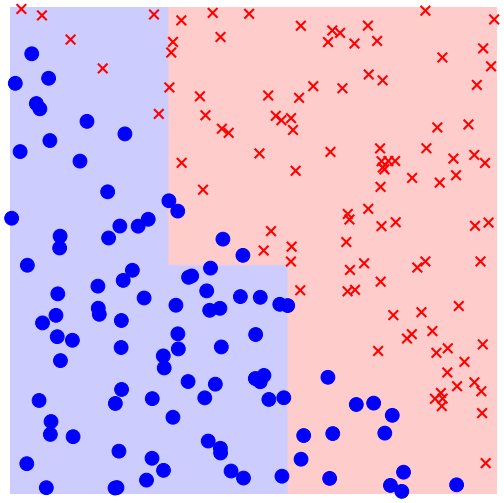
Chosen weak classifier

$\epsilon_3 = 0.2324,\ \alpha_3 = 1.1946$



Re-weight training points

$w_i^{(4)}$**'s**

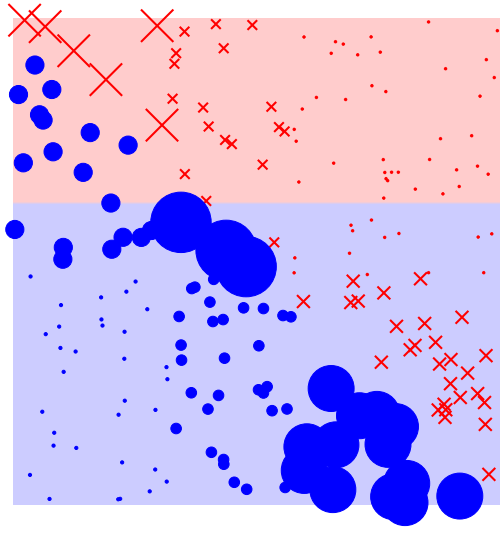


Current strong classifier

$f_3(\mathbf{x})$

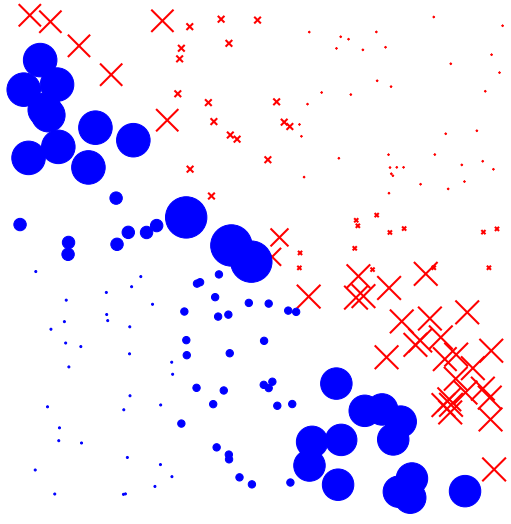
Example

Round 4



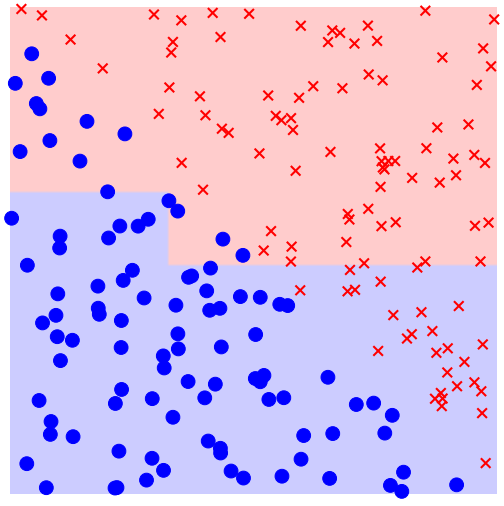
Chosen weak classifier

$\epsilon_4 = 0.2714,\ \alpha_4 = 0.9874$



Re-weight training points

$w_i^{(5)}$**'s**

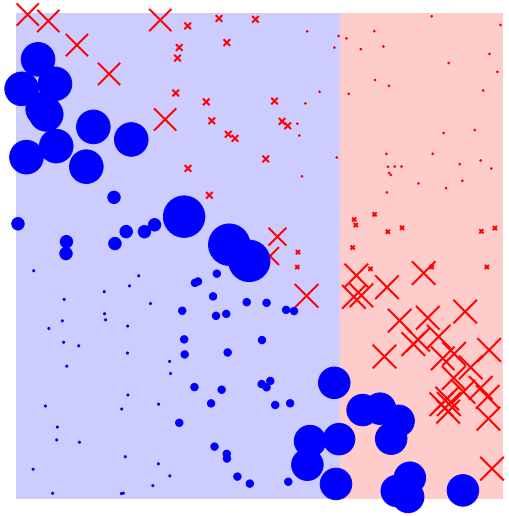


Current strong classifier

$f_4(\mathbf{x})$

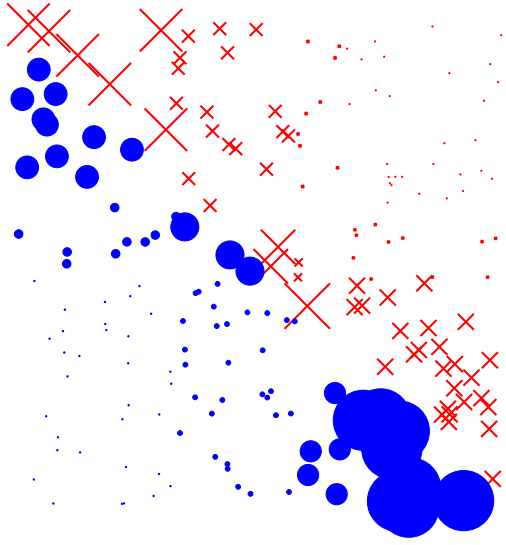
Example

Round 5



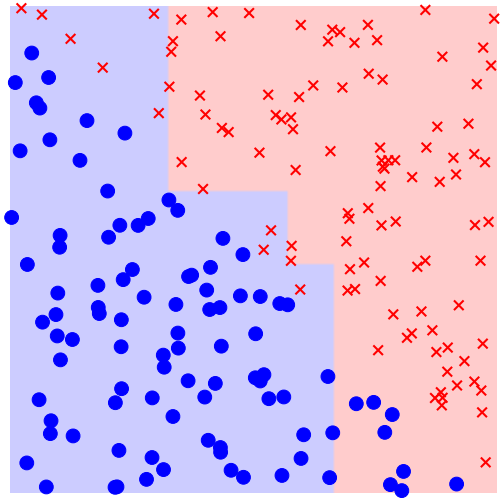
Chosen weak classifier

$\epsilon_5 = 0.2616,\ \alpha_5 = 1.0375$



Re-weight training points

$w_i^{(6)}$**'s**

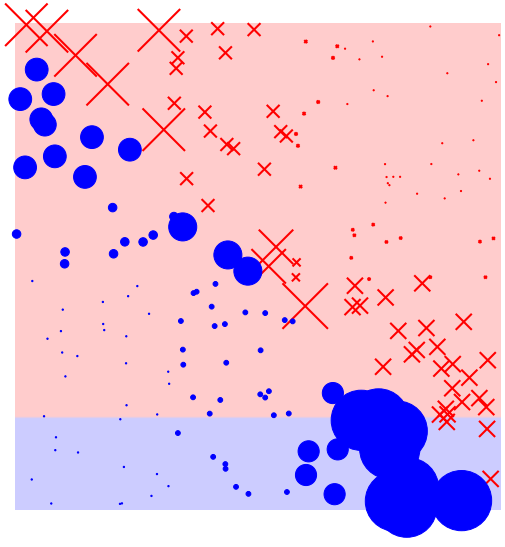


Current strong classifier

$f_5(\mathbf{x})$

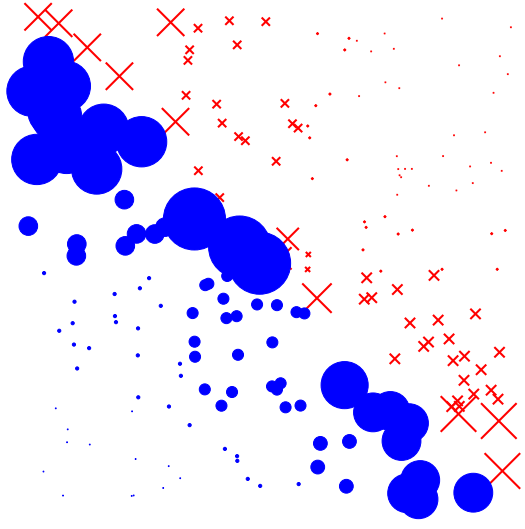
Example

Round 6



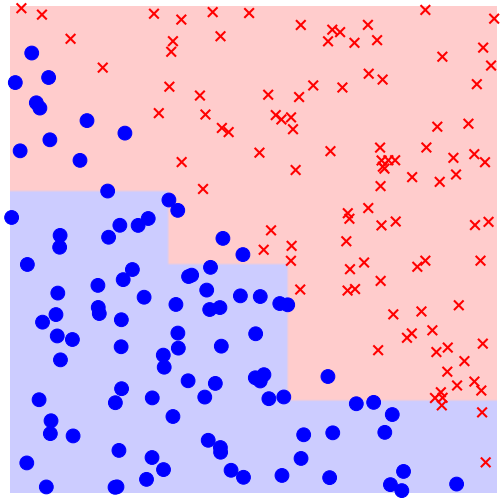
Chosen weak classifier

$\epsilon_6 = 0.2262,\ \alpha_6 = 1.2298$



Re-weight training points

$w_i^{(7)}$**'s**

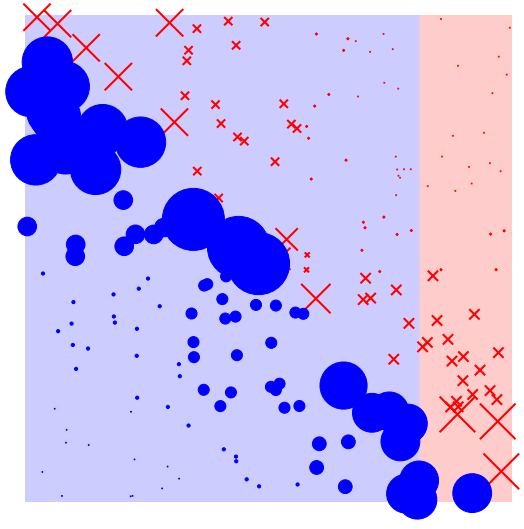


Current strong classifier

$f_6(\mathbf{x})$

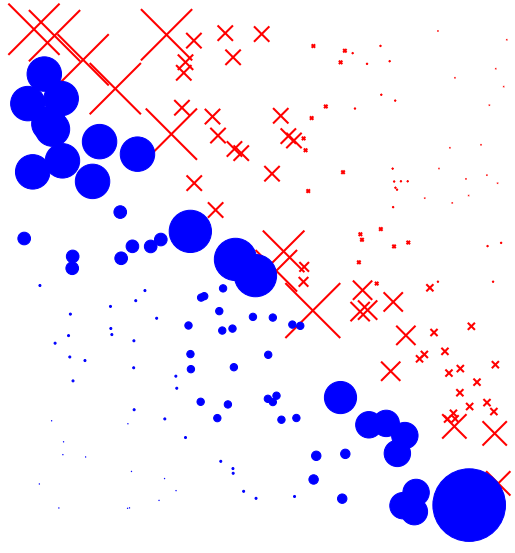
Example

Round 7



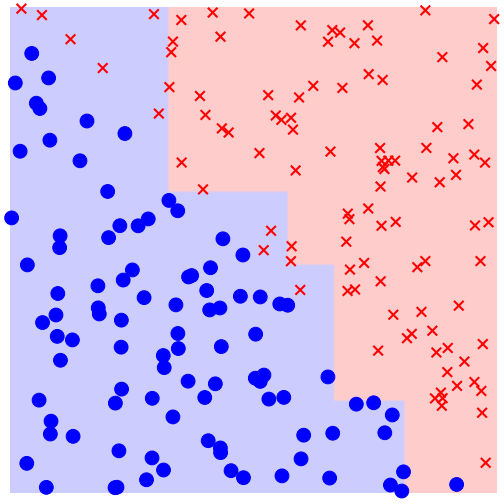
Chosen weak classifier

$\epsilon_7 = 0.2680,\ \alpha_7 = 1.0049$



Re-weight training points

$w_i^{(8)}$**'s**

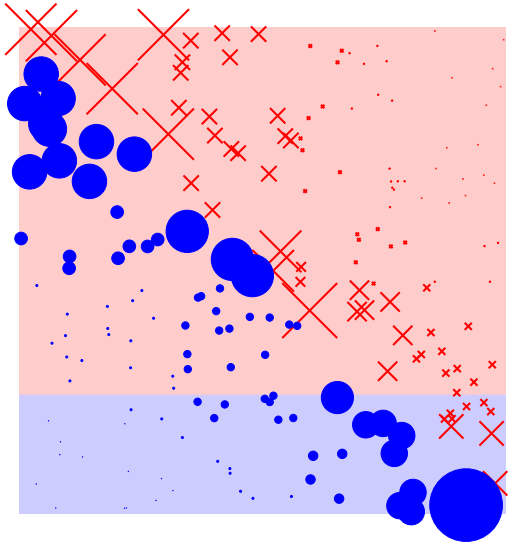


Current strong classifier

$f_7(\mathbf{x})$

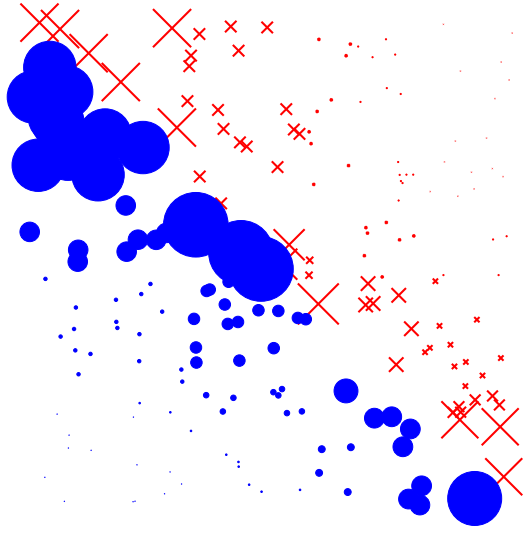
Example

Round 8



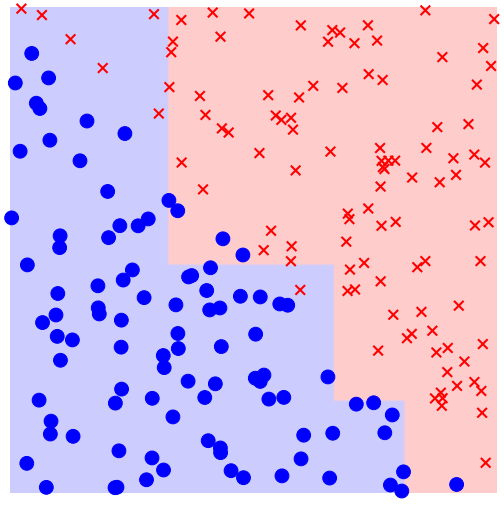
Chosen weak classifier

$\epsilon_8 = 0.3282,\ \alpha_8 = 0.7165$



Re-weight training points

$w_i^{(9)}$**'s**

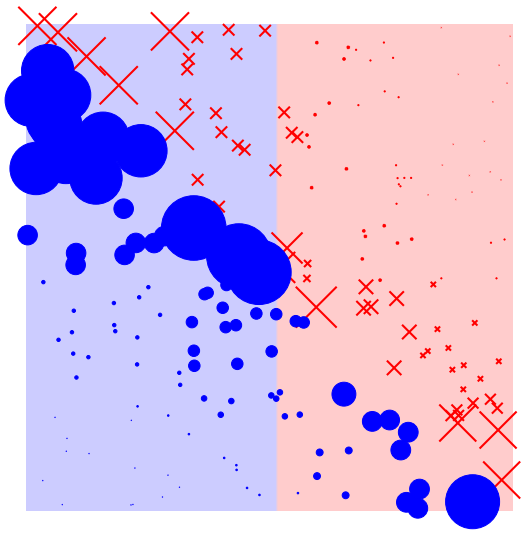


Current strong classifier

$f_8(\mathbf{x})$

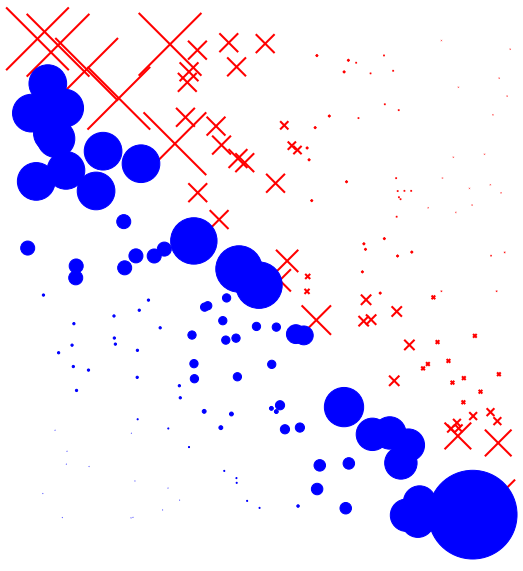
Example

Round 9



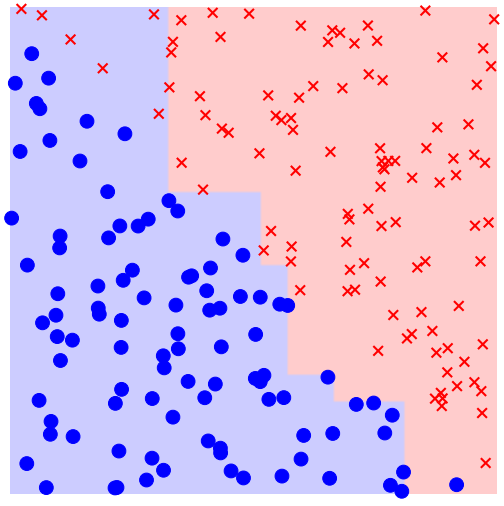
Chosen weak classifier

$\epsilon_9 = 0.3048,\ \alpha_9 = 0.8246$



Re-weight training points

$w_i^{(10)}$**'s**

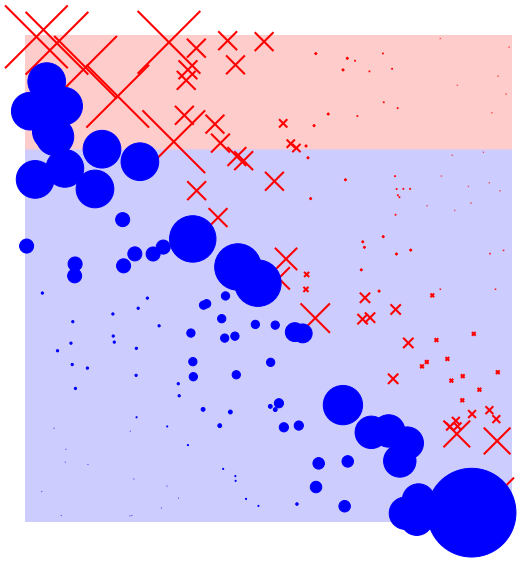


Current strong classifier

$f_9(\mathbf{x})$

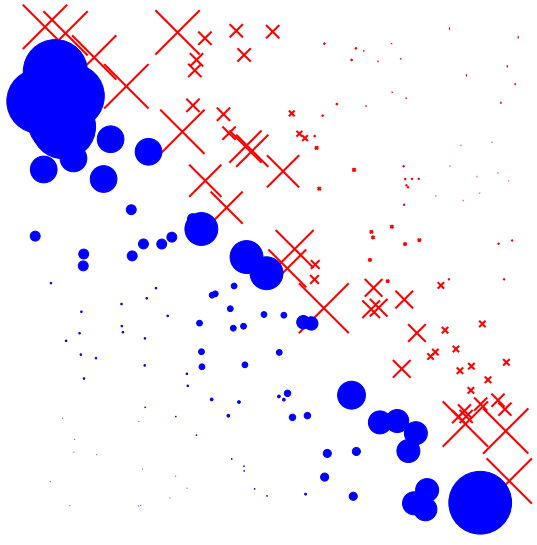
Example

Round 10



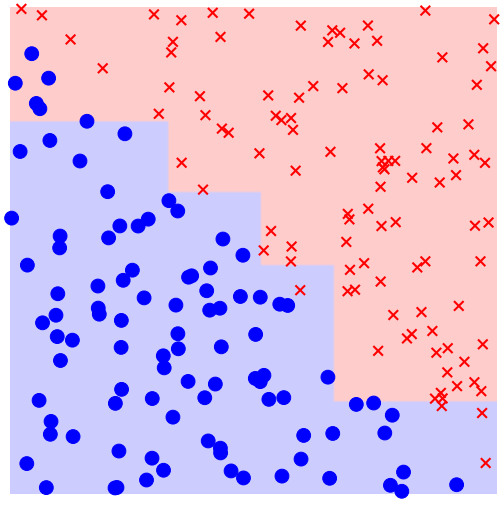
Chosen weak classifier

$\epsilon_{10} = 0.2943,\ \alpha_{10} = 0.8744$



Re-weight training points

$w_i^{(11)}$**'s**

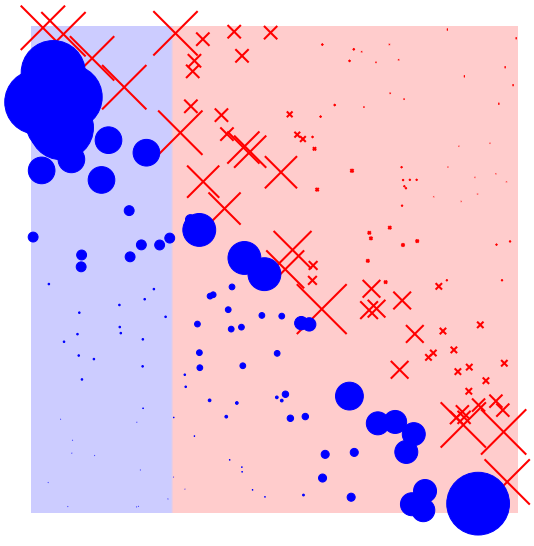


Current strong classifier

$f_{10}(\mathbf{x})$

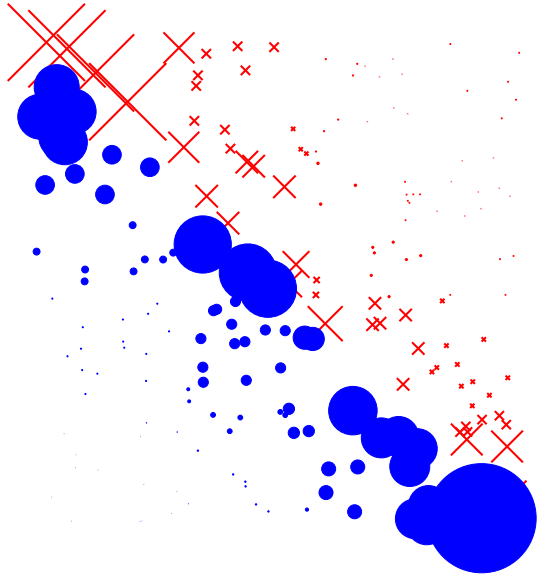
Example

Round 11



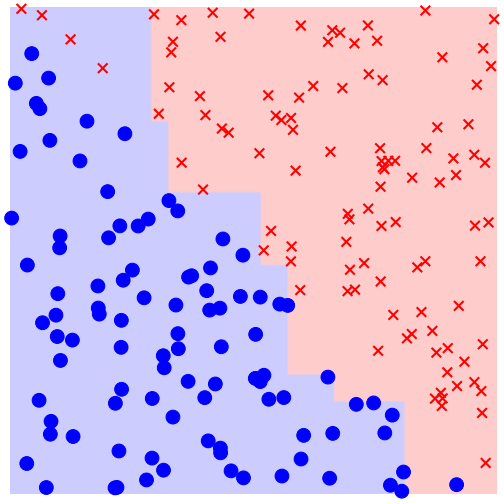
Chosen weak classifier

$\epsilon_{11} = 0.2876$, $\alpha_{11} = 0.9071$



Re-weight training points

$w_i^{(12)}$**'s**



Current strong classifier

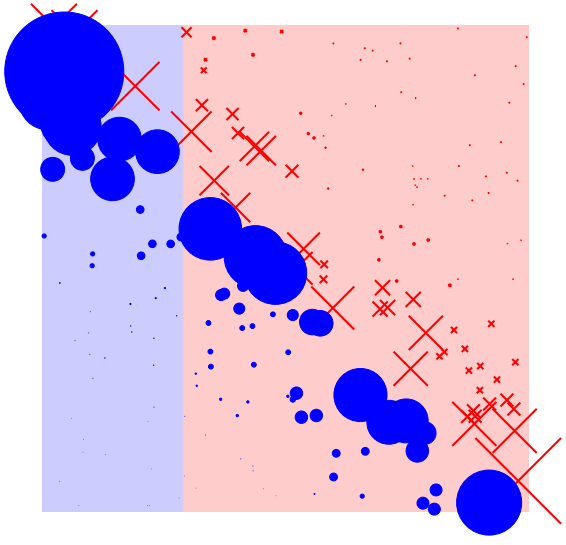
$f_{11}(\mathbf{x})$

Example

.....

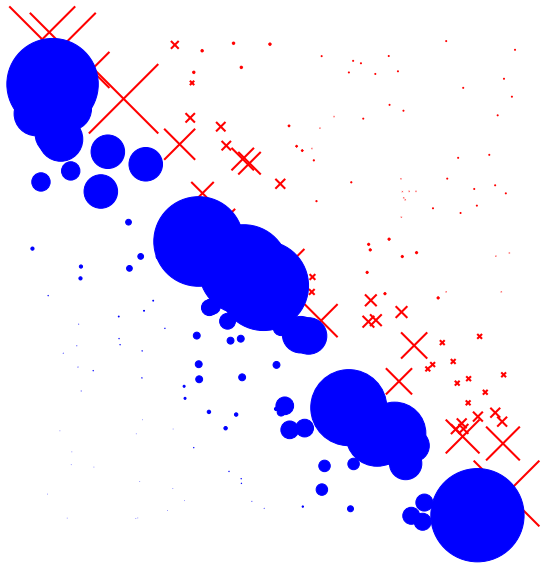
Example

Round 21



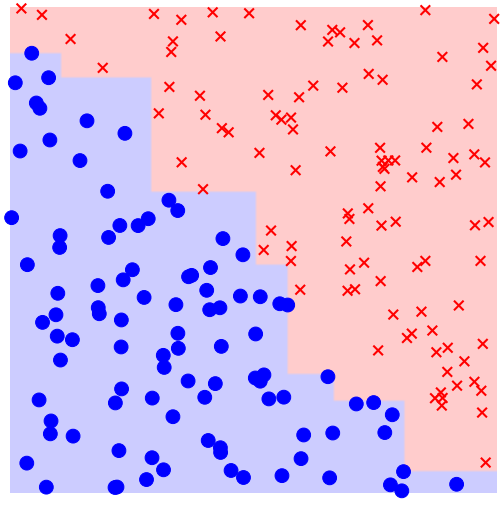
Chosen weak classifier

$\epsilon_{21} = 0.3491,\ \alpha_{21} = 0.6232$



Re-weight training points

$w_i^{(22)}$**'s**



Current strong classifier

$f_{21}(\mathbf{x})$

Boosting Algorithm (Adaboost)

Input:

- Labeled training data

$$\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$$

of inputs $\mathbf{x}_j \in \mathbb{R}^d$ and their labels $y_j \in \{-1, 1\}$.

- A set/class $\mathcal{H}$ of possible weak classifiers.

Output: A strong classifier - weighted sum of weak classifiers:

$$f_T(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right)$$

Boosting Algorithm (Adaboost)

Input:

- Labeled training data

$$\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$$

of inputs $\mathbf{x}_j \in \mathbb{R}^d$ and their labels $y_j \in \{-1, 1\}$.

- A set/class $\mathcal{H}$ of possible weak classifiers.

Output: A strong classifier - weighted sum of weak classifiers:

$$f_T(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right)$$

Boosting Algorithm (Adaboost)

Initialize:

- Introduce a weight, $w_j^{(1)}$, for each training example.
- Set $w_j^{(1)} = \frac{1}{m}$ for each j.

Boosting Algorithm (Adaboost)

Iterate: for $t = 1, \ldots, T$

1. Train classifier $h_t \in \mathcal{H}$ using $\mathcal{S}$ and $w_1^{(t)}, \ldots, w_m^{(t)}$.
2. Compute the training error

$$\epsilon_t = \sum_{j=1}^{m} w_j^{(t)} \text{ Ind}\left(y_j \neq h_t(\mathbf{x}_j)\right)$$

3. If $\epsilon_t \approx .5$ break out of loop.
4. Set $\alpha_t = \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
5. Update weights using:

$$w_j^{(t+1)} = w_j^{(t)} e^{\alpha_t \text{ Ind}(y_j \neq h_t(\mathbf{x}_j))}$$

6. Normalize the weights so that they sum to 1.

Note: $\text{Ind}\,(x) = 1$ if $x =$ TRUE otherwise $\text{Ind}\,(x) = 0$

Properties of the Boosting algorithm

Training Error: Training error $\to 0$ exponentially.

Good Generalization Properties: Would expect over-fitting but even when training error vanishes the test error asymptotes - no-overfitting observed.

Why? One thesis is that Boosting tries to increase the margin of the training examples even when the training error is zero:

$$f_T(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right) = \text{sign}\left(\phi_T(\mathbf{x})\right)$$

Margin of a correctly classified example is: $y_i\, \phi_T(\mathbf{x}_i)$

The larger the margin $\implies$ further example is from the decision boundary $\implies$ better generalization ability.

Properties of the Boosting algorithm

Training Error: Training error $\to 0$ exponentially.

Good Generalization Properties: Would expect over-fitting but even when training error vanishes the test error asymptotes - no-overfitting observed.

Why? One thesis is that Boosting tries to increase the margin of the training examples even when the training error is zero:

$$f_T(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right) = \text{sign}\left(\phi_T(\mathbf{x})\right)$$

Margin of a correctly classified example is: $y_i\, \phi_T(\mathbf{x}_i)$
The larger the margin $\implies$ further example is from the decision boundary $\implies$ better generalization ability.

Properties of the Boosting algorithm

Training Error: Training error $\to 0$ exponentially.

Good Generalization Properties: Would expect over-fitting but even when training error vanishes the test error asymptotes - no-overfitting observed.

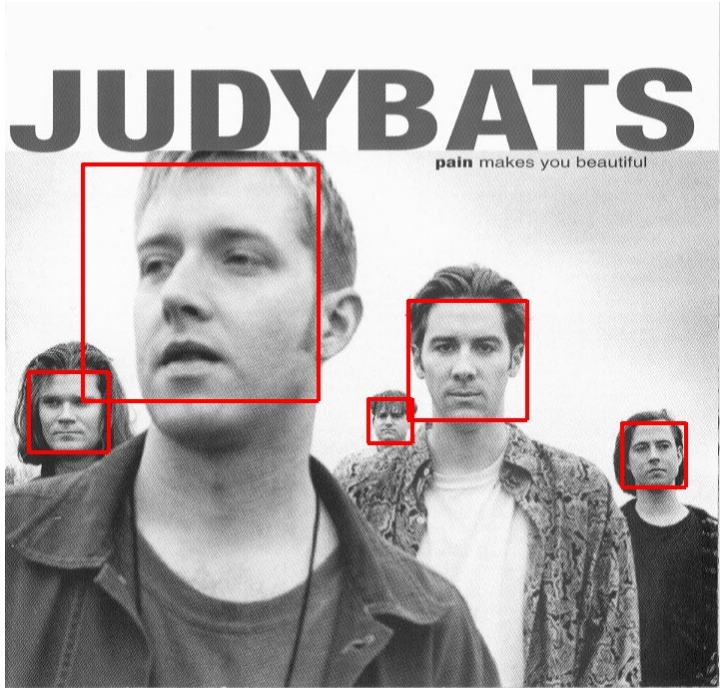
Why? One thesis is that Boosting tries to increase the margin of the training examples even when the training error is zero:

$$f_T(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right) = \text{sign}\left(\phi_T(\mathbf{x})\right)$$

Margin of a correctly classified example is: $y_i\, \phi_T(\mathbf{x}_i)$

The larger the margin $\implies$ further example is from the decision boundary $\implies$ better generalization ability.

Viola & Jones Face Detection



- Most state-of-the-art face detection on mobile phones, digital cameras etc. are based on this algorithm.
- Example of a classifier constructed using the Boosting algorithm.

Viola & Jones: Training data

Positive training examples: Image patches corresponding to faces - $(\mathbf{x}_i, 1)$.

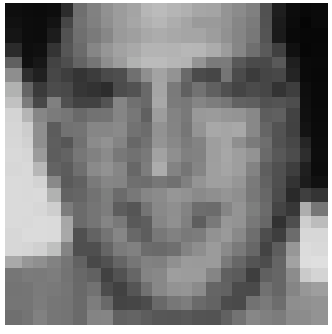
Negative training examples: Random image patches from images not containing faces - $(\mathbf{x}_j, -1)$.

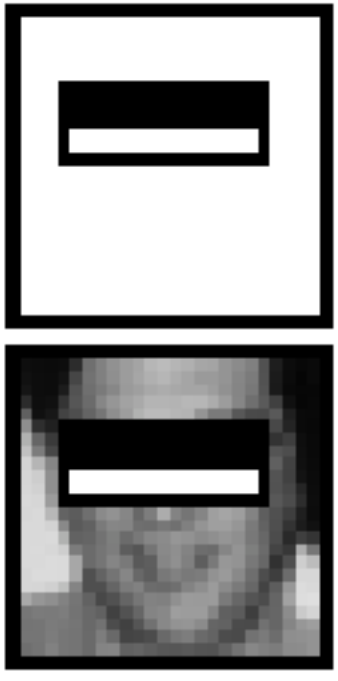
Note: All patches are re-scaled to have same size.

⇑

Positive training examples

Viola & Jones: Weak classifier





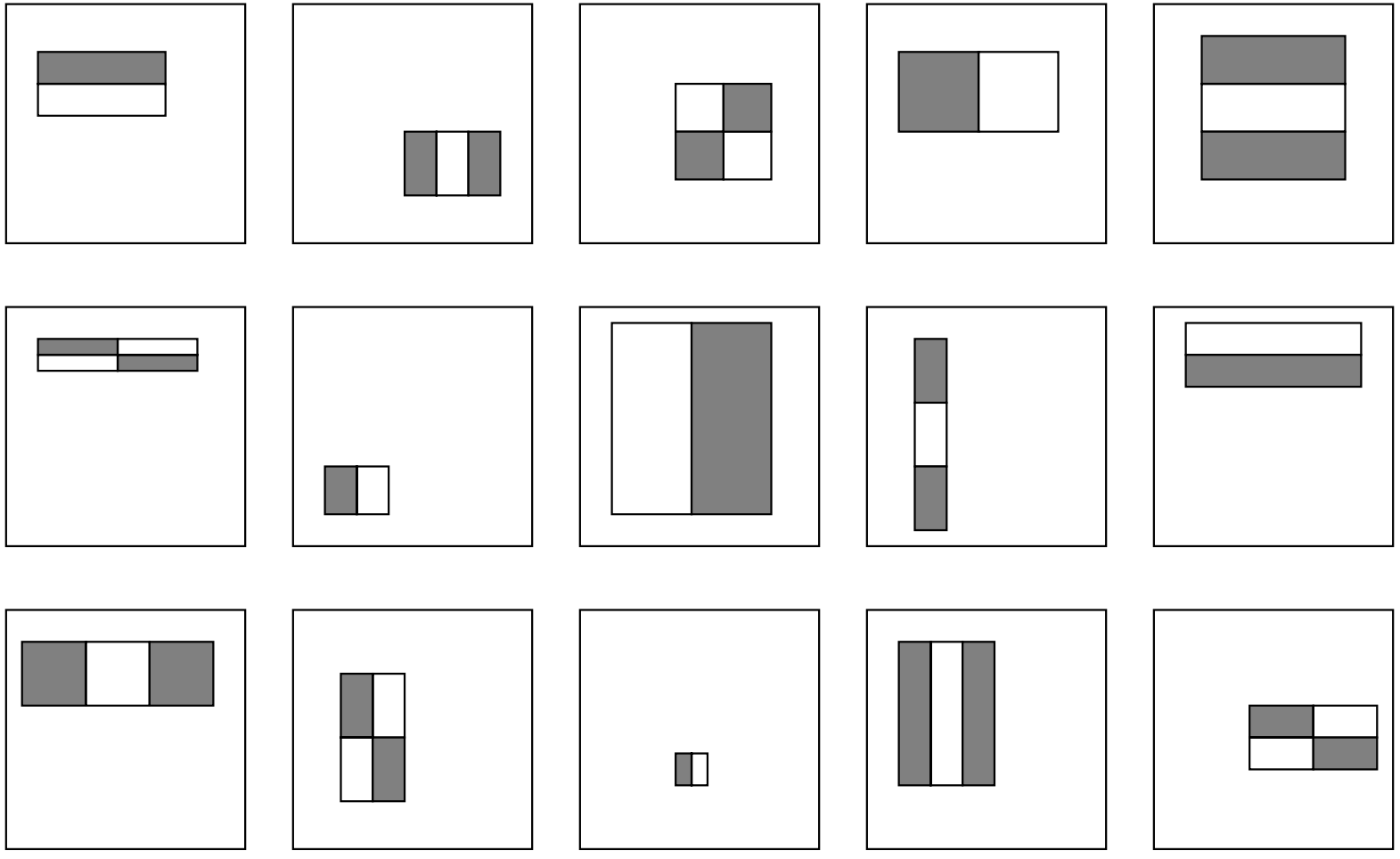
FACE or **NON-FACE**

Input: x **Apply filter:** $f^j(\mathbf{x})$ **Output:** $h(\mathbf{x}) = (f^j(\mathbf{x}) > \theta)$

Filters used compute differences between sums of pixels in adjacent rectangles. These can be computed very quickly using **Integral Images**.

Viola & Jones: Filters Considered

Huge **library** of possible filters, $f^1, \ldots, f^d$ with $d \approx 16,000,000$.



Viola & Jones: AdaBoost

Recap: define weak classifier as

$$h_t(\mathbf{x}) = \begin{cases} 1 & \text{if } f^{j_t}(\mathbf{x}) > \theta_t \\ -1 & \text{otherwise} \end{cases}$$

Use AdaBoost to efficiently choose the **best weak classifiers** and to **combine** them.

Remember: a weak classifier corresponds to a filter type and a threshold.

Viola & Jones: AdaBoost

For $t = 1, \ldots, T$

- for each filter type j
  1. Apply filter, f^j, to each example.
  2. Sort examples by their filter responses.
  3. Select best threshold for this classifier: θ_{tj}.
  4. Keep record of error of this classifier: ϵ_{tj}.
- Select the filter-threshold combination (weak classifier j^*) with minimum error. Then set $j_t = j^*$, $\epsilon_t = \epsilon_{tj^*}$ and $\theta_t = \theta_{tj^*}$.
- Re-weight examples according to the AdaBoost formualae.

Note: (There are many tricks to make this implementation more efficient.)

Viola & Jones: Sliding window

Remember: Better classification rates if use a classifier, f_T, with large T.

Given an new image, I, detect the faces in the image by:

- for each plausible face size s
  - for each possible patch centre c
    1. Extract sub-patch of size s at c from I.
    2. Re-scale patch to size of training patches.
    3. Apply detector to patch.
    4. Keep record of s and c if the detector returns positive.

This is a **lot** of patches to be examined. If T is very large processing an image will be very slow!

Viola & Jones: Sliding window

Remember: Better classification rates if use a classifier, f_T, with large T.

Given an new image, I, detect the faces in the image by:

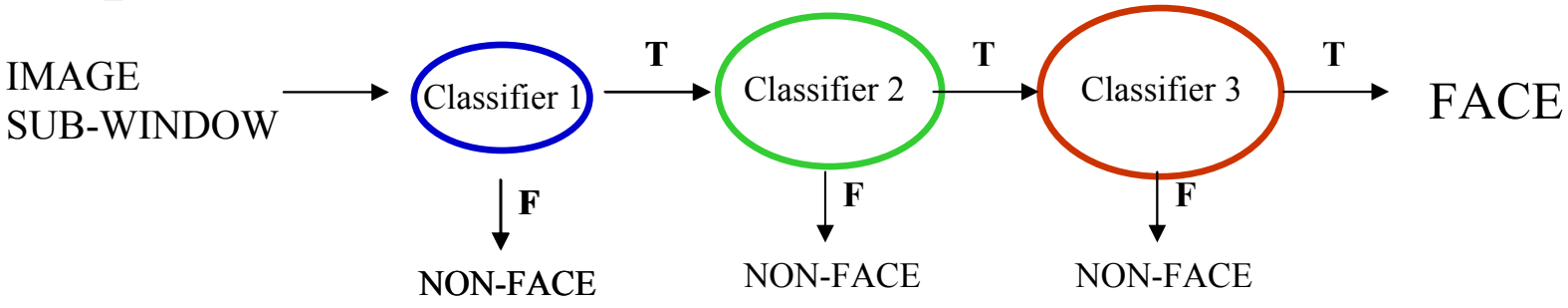
- for each plausible face size s
  - for each possible patch centre c
    1. Extract sub-patch of size s at c from I.
    2. Re-scale patch to size of training patches.
    3. Apply detector to patch.
    4. Keep record of s and c if the detector returns positive.

This is a **lot** of patches to be examined. If T is very large processing an image will be very slow!

Viola & Jones: Cascade of classifiers

But:
only a tiny proportion of the patches will be faces **and** many of them will not look anything like a face.

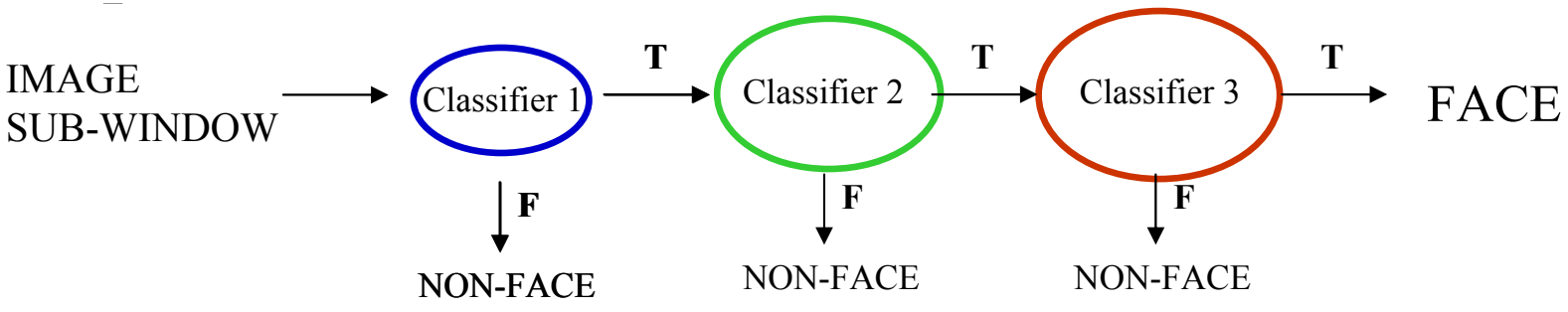
Exploit this fact: Introduce a cascade of increasingly strong classifiers



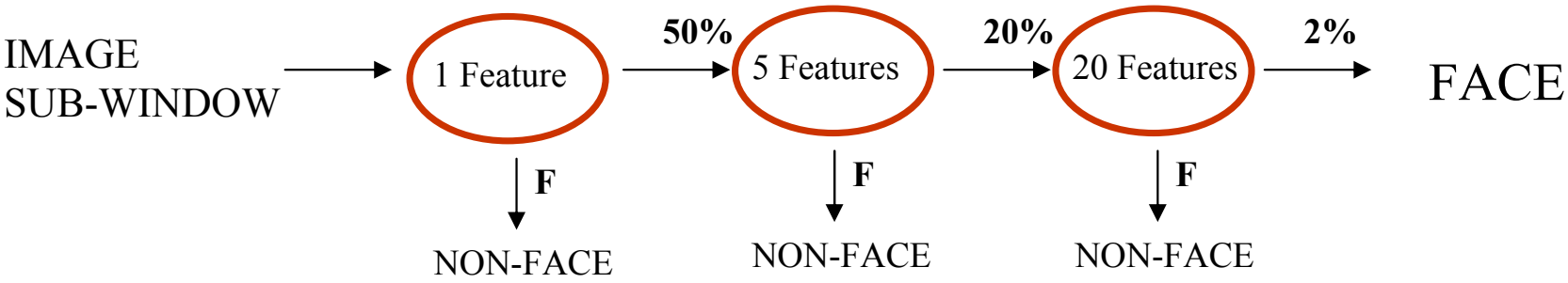
Viola & Jones: Cascade of classifiers

But:
only a tiny proportion of the patches will be faces **and** many of them will not look anything like a face.

Exploit this fact: Introduce a cascade of increasingly strong classifiers

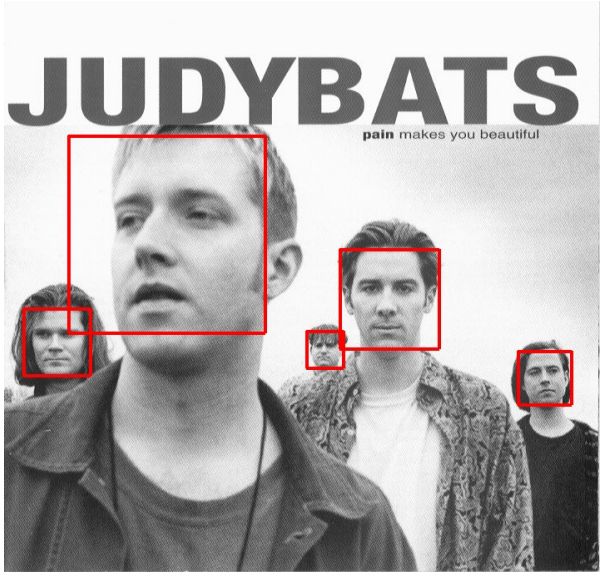


Viola & Jones: Cascade of classifiers



- A 1 feature classifier achieves 100% detection rate and about 50% false positive rate.
- A 5 feature classifier achieves 100% detection rate and 40% false positive rate (20% cumulative) - using data from previous stage.
- A 20 feature classifier achieves 100% detection rate with 10% false positive rate (2% cumulative).

Viola & Jones: Typical Results



Ensemble method: **Bagging**

High variance, Low bias classifiers

Bias of a classifier is the squared discrepancy between the averaged estimated and true function

$$(E[\hat{f}(\mathbf{x})] - E[f(\mathbf{x})])^2$$

Variance of a classifier is the expected divergence of the estimated function from its average value:

$$E[(\hat{f}(\mathbf{x}) - E[\hat{f}(\mathbf{x})])^2]$$

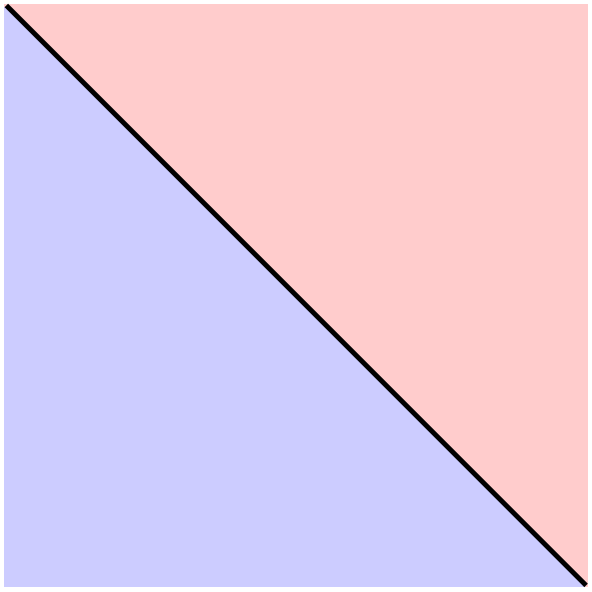
High variance, Low bias classifiers

High variance classifiers - *decision trees* - produce differing decision boundaries which are highly dependent on the training data.

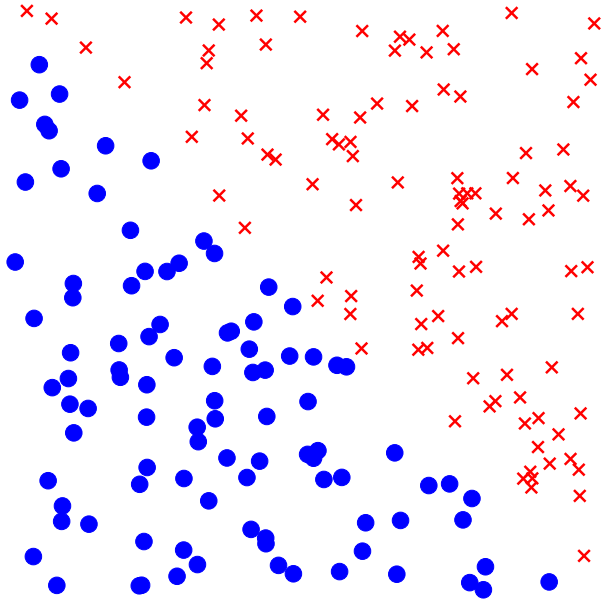
Low bias classifiers - *decision trees* - produce decision boundaries which on average are good approximations to the true decision boundary.

High variance, Low bias classifiers

Binary classification example



True decision boundary

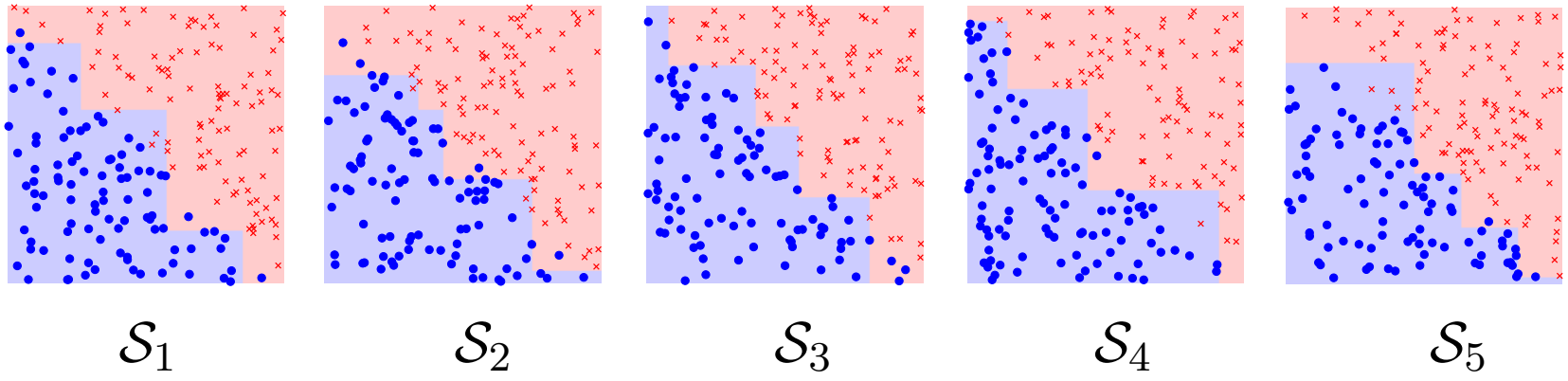


Training data set $\mathcal{S}_i$

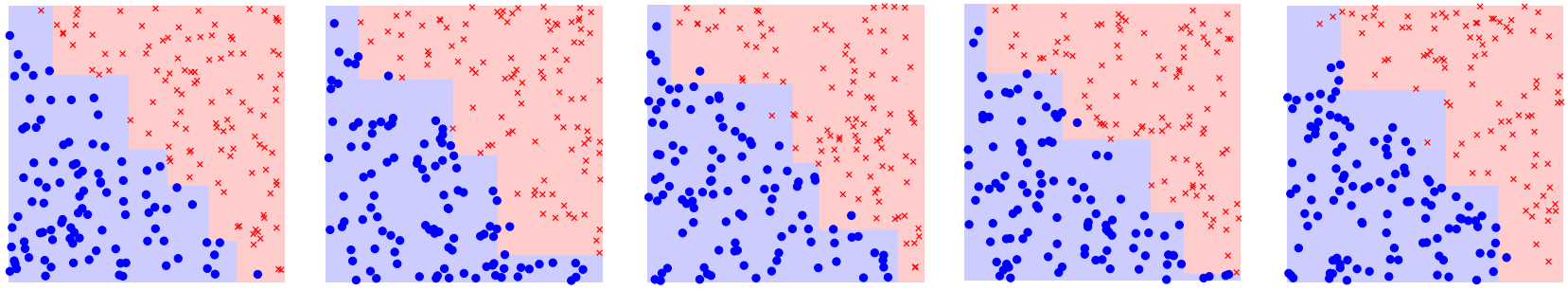
Estimate the true decision boundary with a *decision tree* trained from some labeled training set $\mathcal{S}_i$.

High variance, Low bias classifiers

Estimated decision boundaries found using:



$\mathcal{S}_1$ $\mathcal{S}_2$ $\mathcal{S}_3$ $\mathcal{S}_4$ $\mathcal{S}_5$



$\mathcal{S}_6$ $\mathcal{S}_7$ $\mathcal{S}_8$ $\mathcal{S}_9$ $\mathcal{S}_{10}$

High variance, Low bias classifiers

See how the decision boundaries on the previous slide differ from the

expected decision boundary of the decision tree classifier (with $m = 200$ training points).

High variance, Low bias classifiers

Bagging is a procedure to reduce the variance of our classifier when labelled training data is limited.

Bias of **bagged classifier** may be marginally less than the base classifiers.

Note: it only produces good results for **high variance**, **low bias** classifiers.

Bagging - Bootstrap Aggregating

Input: Training data

$$\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$$

of inputs $\mathbf{x}_j \in \mathbb{R}^d$ and their labels or real values y_j.

Bagging - Bootstrap Aggregating

Iterate: for $b = 1, \ldots, B$

1. Sample training examples, *with replacement*, m times from $\mathcal{S}$ to create $\mathcal{S}_b$.
2. Use this bootstrap sample $\mathcal{S}_b$ to estimate the regression or classification function f_b.

Bagging - Bootstrap Aggregating

Output: The bagging estimate for

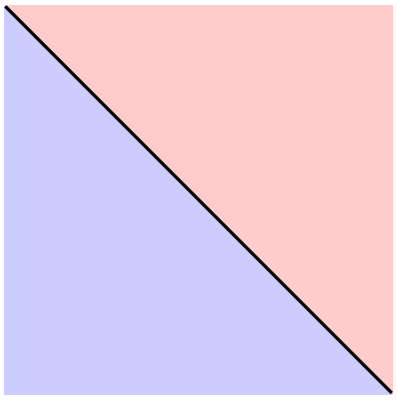
Regression:

$$f_{\text{bag}}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} f_b(\mathbf{x})$$

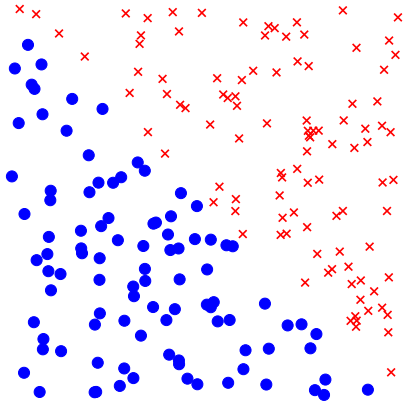
Classification:

$$f_{\text{bag}}(\mathbf{x}) = \arg\max_{1\leq k\leq K} \sum_{b=1}^{B} \text{Ind}\left(f_b(\mathbf{x}) = k\right)$$

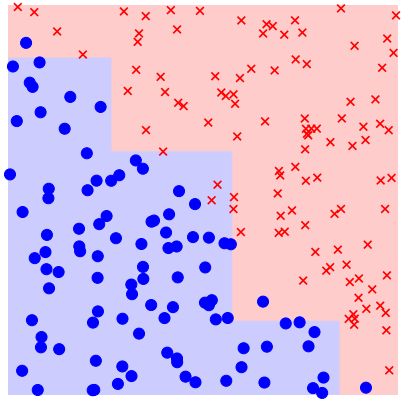
Apply bagging to the original example



Ground truth



$\mathcal{S}_1$

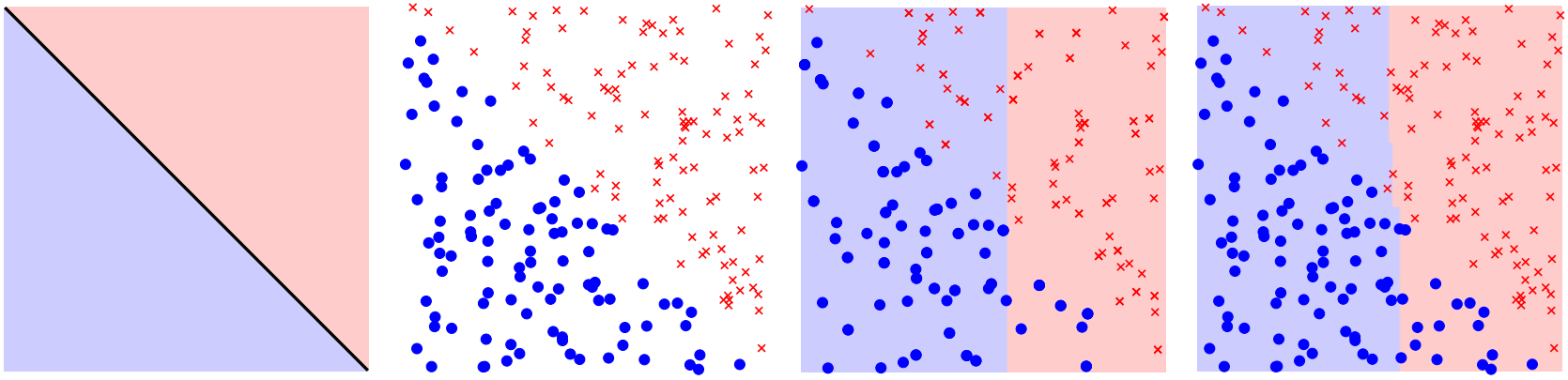


Classifier from $\mathcal{S}_1$

Bagged classifier $B = 100$**.**

Apply bagging to the original example

If we bag a **high bias, low variance** classifier - *oriented horizontal and vertical lines* - we don't get any benefit.



Ground truth

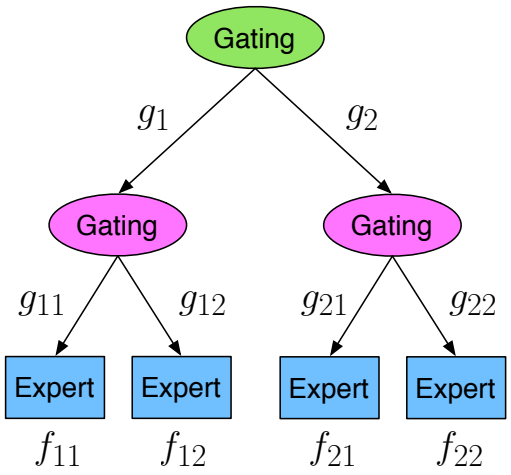
$\mathcal{S}_1$

Classifier from $\mathcal{S}_1$

Bagged classifier $B = 100$.

Ensemble method: **Hierarchical Mixture of Experts**

Hierarchical Mixture of Experts



The function in the figure has the form

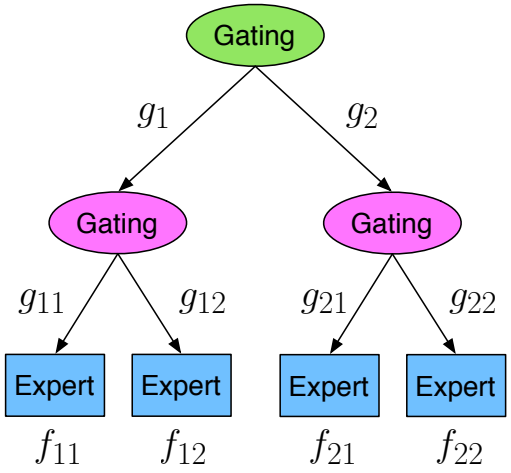
$$f_{\text{HME}}(\mathbf{x}) = \sum_{j=1}^{2} g_j(\mathbf{x}, \mathbf{v}_j) \sum_{l=1}^{2} g_{jl}(\mathbf{x}, \mathbf{v}_{jl}) f_{jl}(\mathbf{x}, \mathbf{w}_{jl})$$

where for classification

$$g_j(\mathbf{x}, \mathbf{v}_j) = \frac{e^{\mathbf{v}_j^t \mathbf{x}}}{\sum_{k=1}^{2} e^{\mathbf{v}_k^t \mathbf{x}}}, \qquad f_{jl}(\mathbf{x}, \mathbf{w}_{jl}) = \frac{1}{1 + e^{-\mathbf{w}_{jl}^t \mathbf{x}}}$$

The parameters $\mathbf{v}_j$'s, $\mathbf{v}_{jl}$'s and $\mathbf{w}_{jl}$'s are learned from the training data using the EM algorithm.

Hierarchical Mixture of Experts



The function in the figure has the form

$$f_{\text{HME}}(\mathbf{x}) = \sum_{j=1}^{2} g_j(\mathbf{x}, \mathbf{v}_j) \sum_{l=1}^{2} g_{jl}(\mathbf{x}, \mathbf{v}_{jl}) f_{jl}(\mathbf{x}, \mathbf{w}_{jl})$$

This algorithms ends up learning

- different classifiers f_{jl} for different parts of the input space
- the soft assignment of each $\mathbf{x}$ to one of these regions via the g_{jl}'s and g_j's

Summary

Summary: Ensemble Prediction

- Can combine many weak classifiers/regressors into a stronger classifier.
  - Voting
  - Averaging
  - Bagging
  - Wisdom of crowds
- If weak classifiers/regressors are better than random.
- If there is sufficient de-correlation (independence) amongst the weak classifiers/regressors.

Summary: Ensemble Prediction & Learning

Can combine many (high-bias) weak classifiers/regressors into a **strong** classifier

- If weak classifiers/regressors are **chosen** and **combined** using knowledge of how well they and others performed on the task on training data.
  - Boosting
  - Hierarchical Mixture of Experts
- The selection and combination encourages the weak classifiers to be complementary, diverse and de-correlated.

Summary: Ensemble Prediction & Learning

Want to know more:

Reading list in Marsland Chapter 7 pages 164-165

P. Viola, M. J. Jones, **Robust real-time face detection**.
International Journal of Computer Vision 57(2): 137-154, 2004.

Next Lecture: Probability Based Learning
Reading: Marsland Chapter 8.1