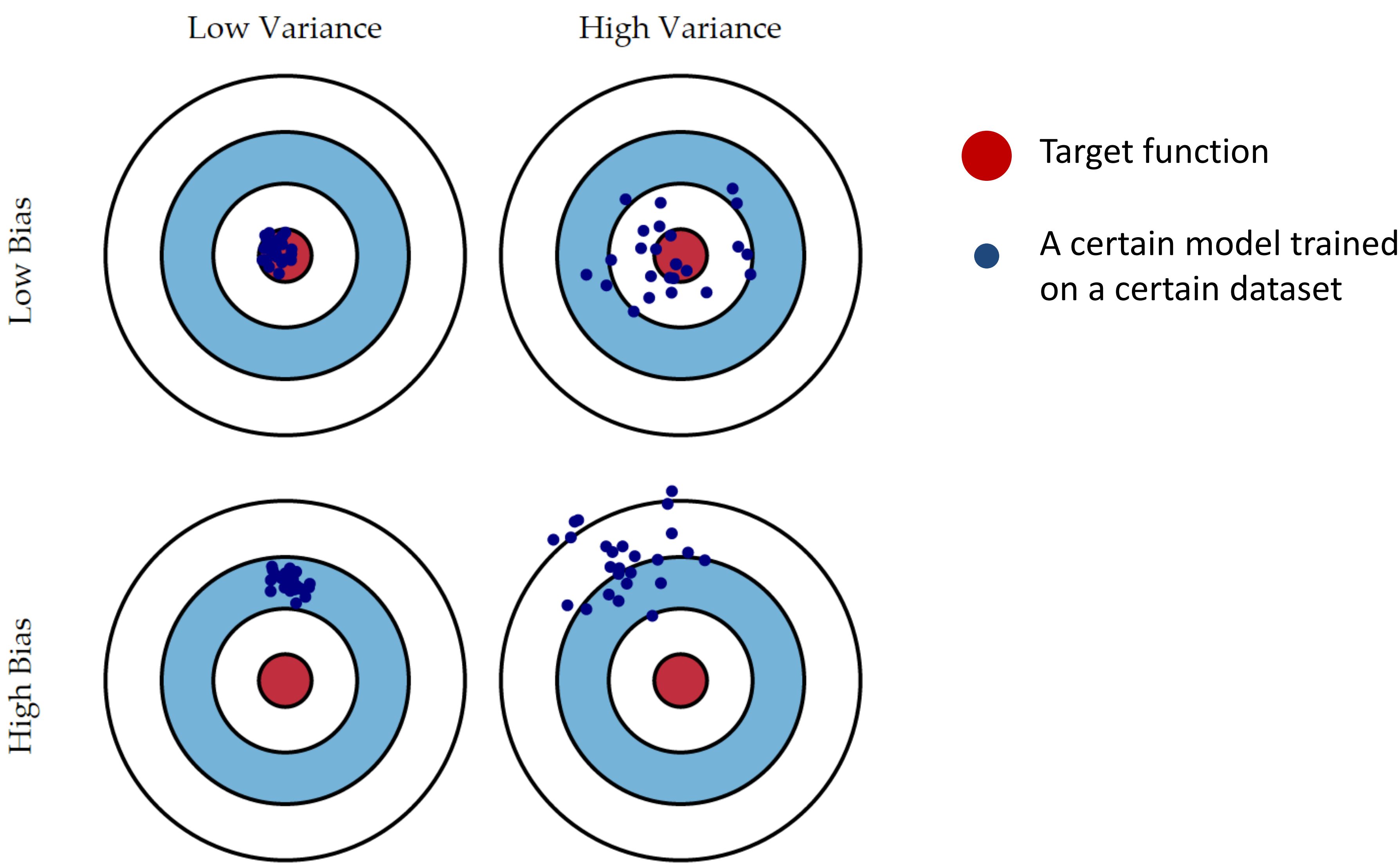
COMP90051 **Statistical Machine Learning**

Semester 2, 2015

Ensemble Learning

THE UNIVERSITY OF MELBOURNE

Bias vs Variance



Source: http://scott.fortmann-roe.com/docs/BiasVariance.html

Ensemble Learning

- Combined models for regression and classification
- Train a set of classifiers instead of a single classifier
- Reduce variance: results are less dependent on peculiarities of a single training set
- Reduce bias: a combination of multiple classifiers may learn a more expressive concept class than a single classifier
- Generally, more diversity → more accurate

Bagging

- Instance manipulation: data resampling using bootstrap.

Model Generation

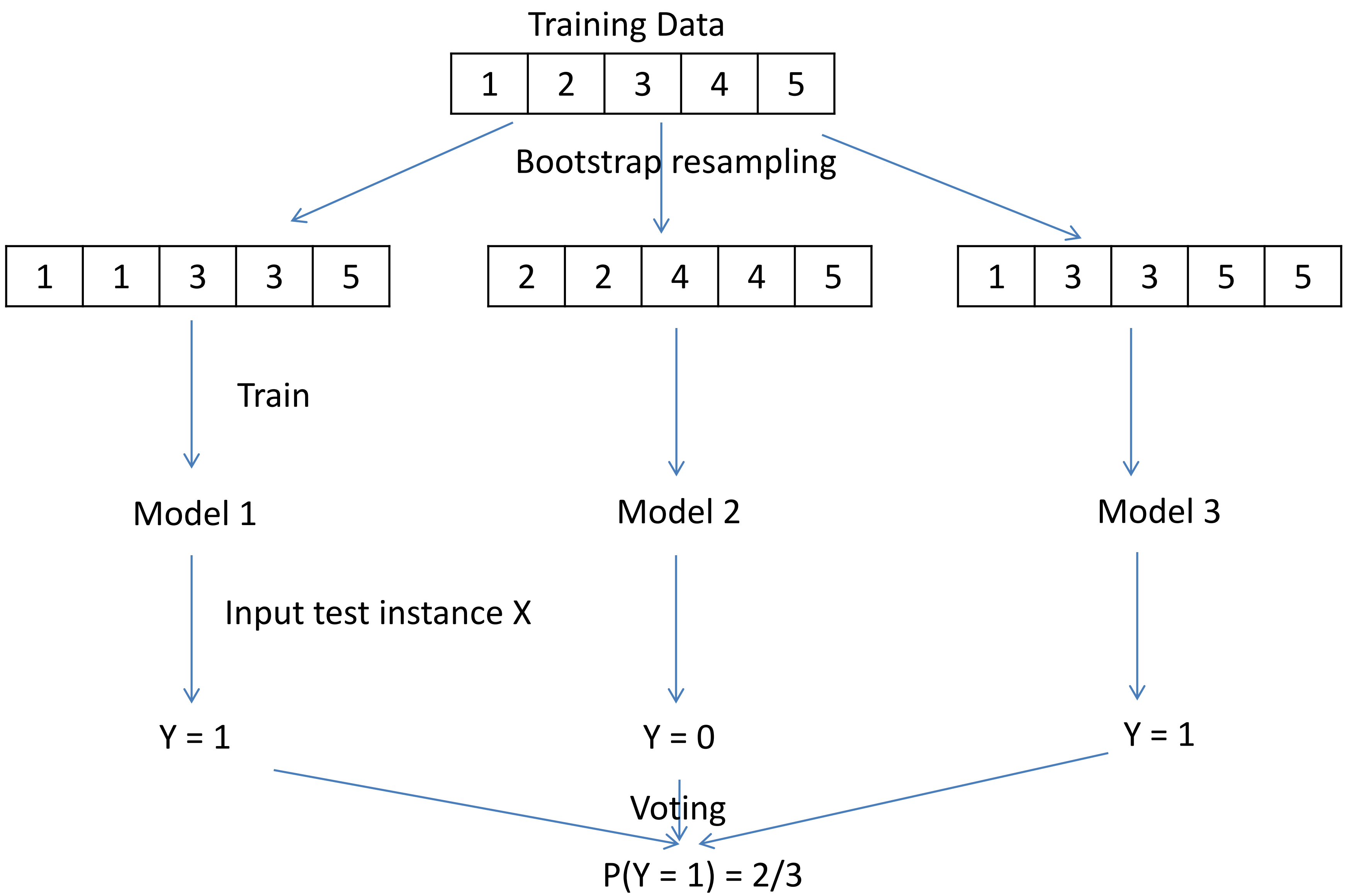
```
Let n be the number of instances in the training data.
For each of t iterations:
  Sample n instances with replacement from training data.
  Apply the learning algorithm to the sample.
  Store the resulting model.
```

Classification

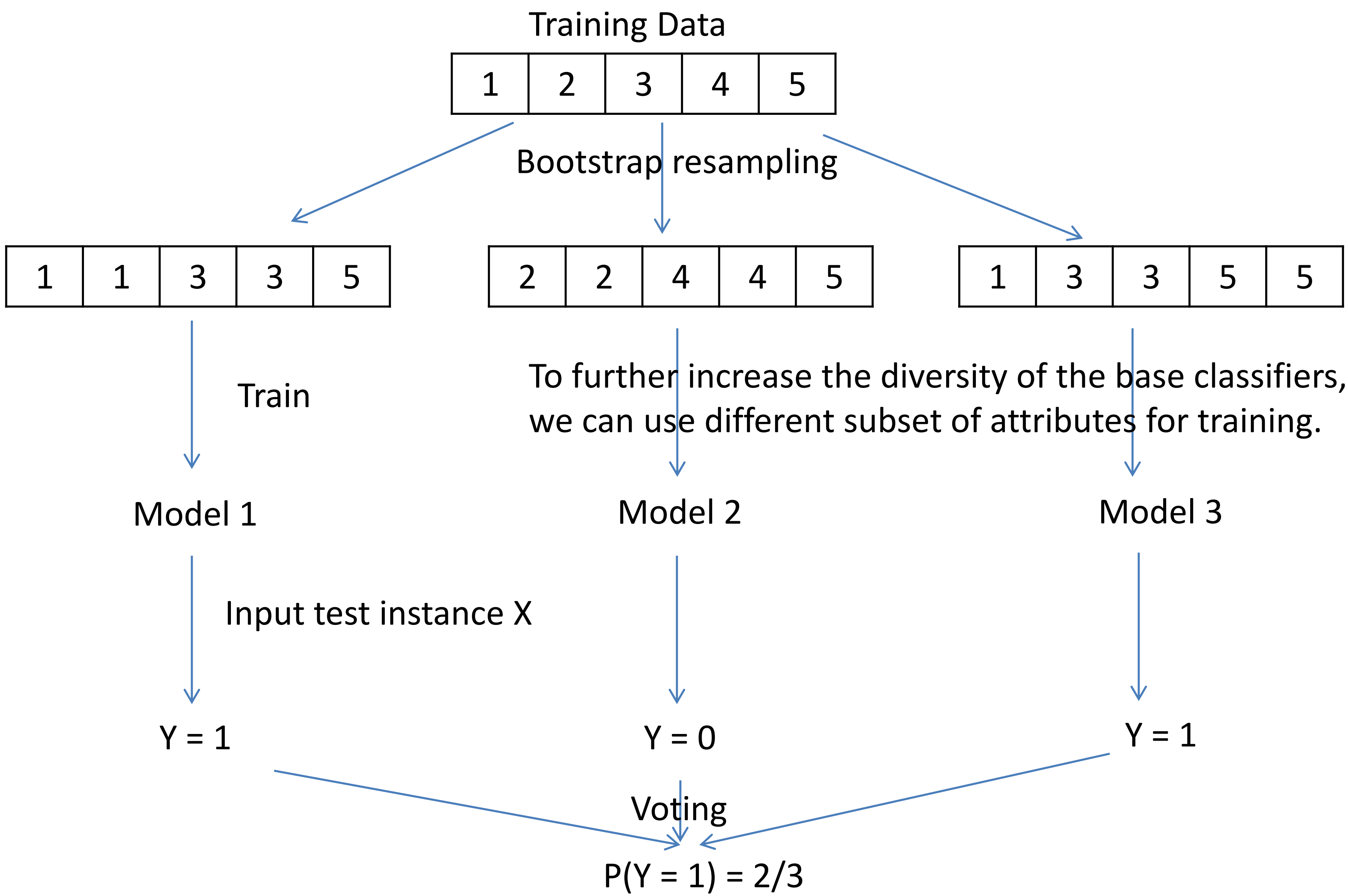
```
For each of the t models:
  Predict class of instance using model.
Return class that has been predicted most often.
```


Source: Witten, Ian H. et al.. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2011.

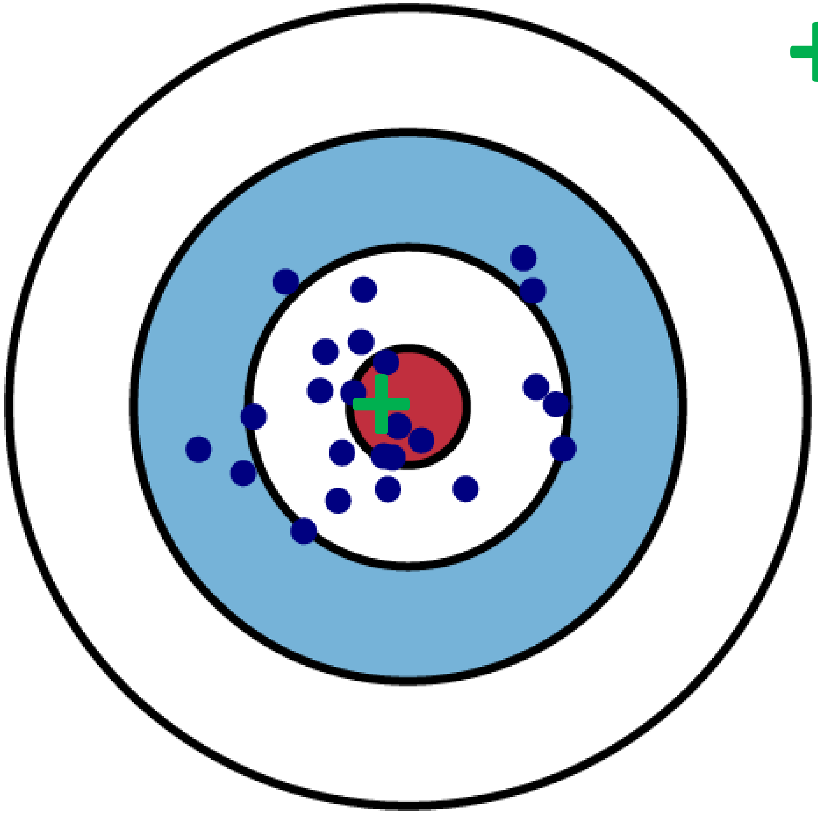
Bagging


Training Data
1 2 3 4 5
Bootstrap resampling
1 1 3 3 5
2 2 4 4 5
1 3 3 5 5
Train
Model 1
Model 2
Model 3
Input test instance X
Y = 1
Y = 0
Y = 1
Voting
P(Y = 1) = 2/3

Bagging


Training Data
1 2 3 4 5
Bootstrap resampling
1 1 3 3 5
2 2 4 4 5
1 3 3 5 5
Train
To further increase the diversity of the base classifiers, we can use different subset of attributes for training.
Model 1
Model 2
Model 3
Input test instance X
Y = 1
Y = 0
Y = 1
Voting
P(Y = 1) = 2/3

Effect on Variance

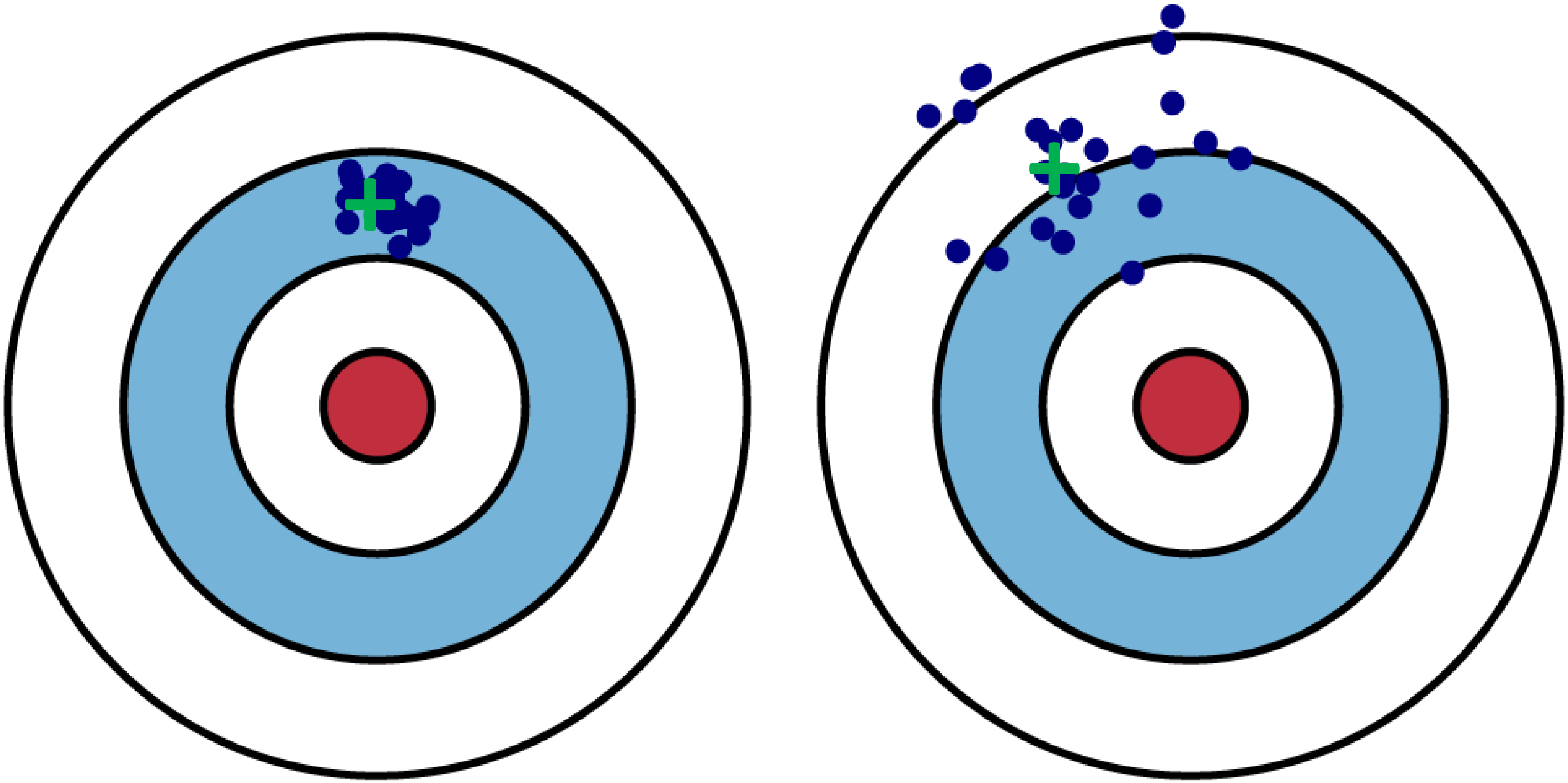


+ : average

● A base model

Reducing variance via averaging

Effect on Bias



Bagging: Resampling

- Bagging reduces variance by averaging
- Bagging has little effect on bias
  - BUT, it generally won’t cause bias.
- Each base classifier is trained on less real data
- Works better with unstable classifiers

Boosting

- Require classifiers that can handle weighted instances
  - E.g. C4.5 fractional instances
- “hard” instances have higher weights.
- In Bagging, models are built separately.
- In Boosting, models are built iterative.

AdaBoost

Model Generation

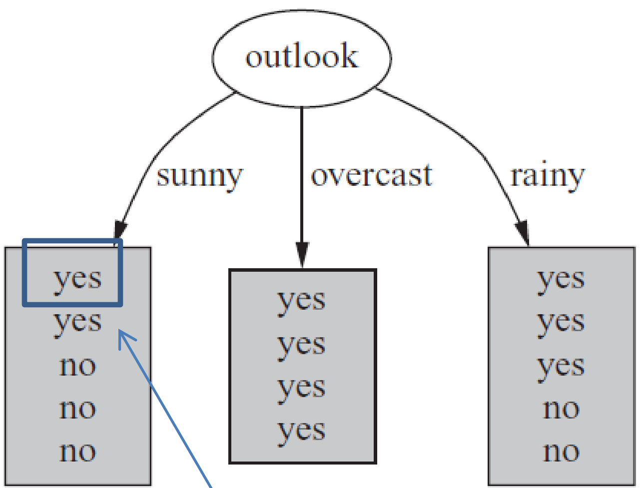
```
Assign equal weight to each training instance.
For each of t iterations:
  Apply learning algorithm to weighted dataset and store resulting
    model.
  Compute error e of model on weighted dataset and store error.
  If e equal to zero, or e greater or equal to 0.5:
    Terminate model generation.
  For each instance in dataset:
    If instance classified correctly by model:
       Multiply weight of instance by e / (1 - e).
  Normalize weight of all instances.
```

Classification

```
Assign weight of zero to all classes.
For each of the t (or less) models:
  Add -log(e / (1 - e)) to weight of class predicted by model.
Return class with highest weight.
```


Source: Witten, Ian H. et al.. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2011.

Instances' Weight in C4.5



- Entropy for each branch
  - $Info([2,3]) = 0.971 bits$
  - $Info([4,0]) = 0\ bits$
  - $Info([3,2]) = 0.971 bits$
- Average entropy:
  - $Info([2,3],[4,0],[3,2]) = \boxed{\frac{5}{14}} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971 = 0.693\ bits$

The weight for current branch

Not necessary to be an integer, could be fractional number: e.g. "1.5 yes" or "0.8 yes"

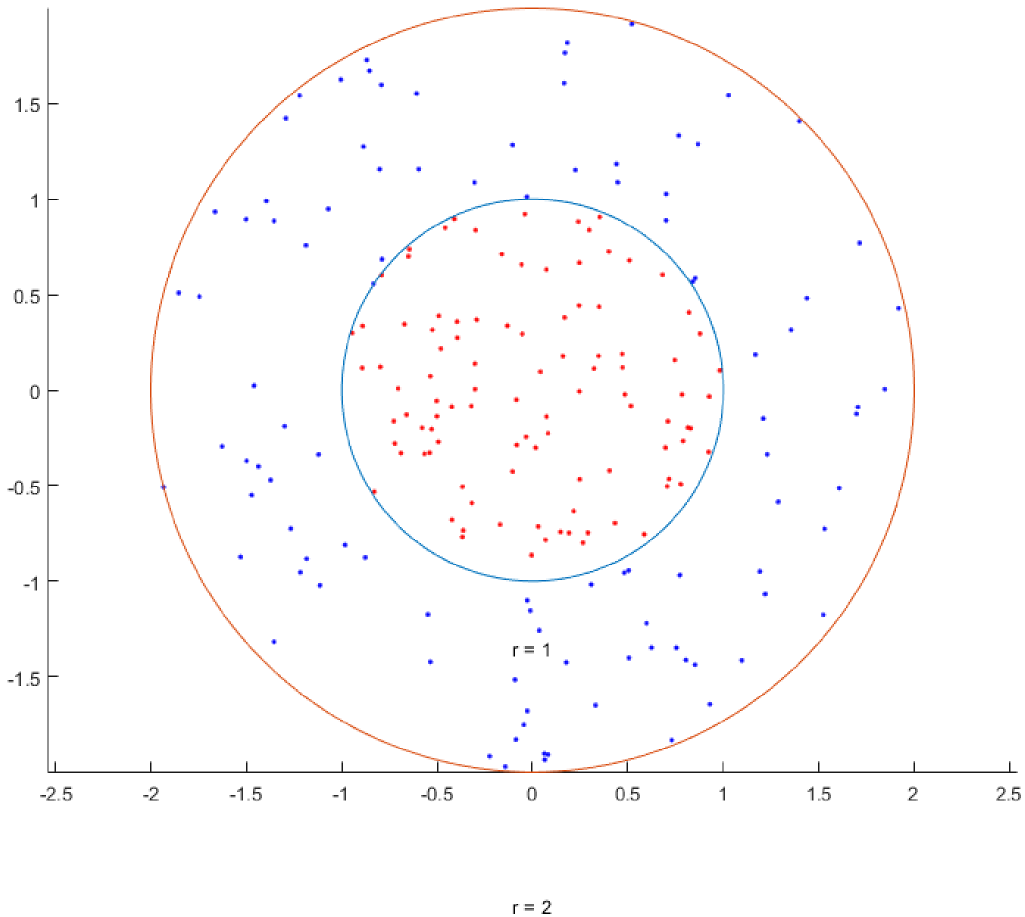
Source: Witten, Ian H. et al.. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2011.

AdaBoost

- A Matlab demo, can be found on LMS
  - Click and play
  - `weakLearnerNum`: how many weak classifiers will learn during AdaBoost training

Source: Wang Kang, http://staff.ustc.edu.cn/~xjchen99

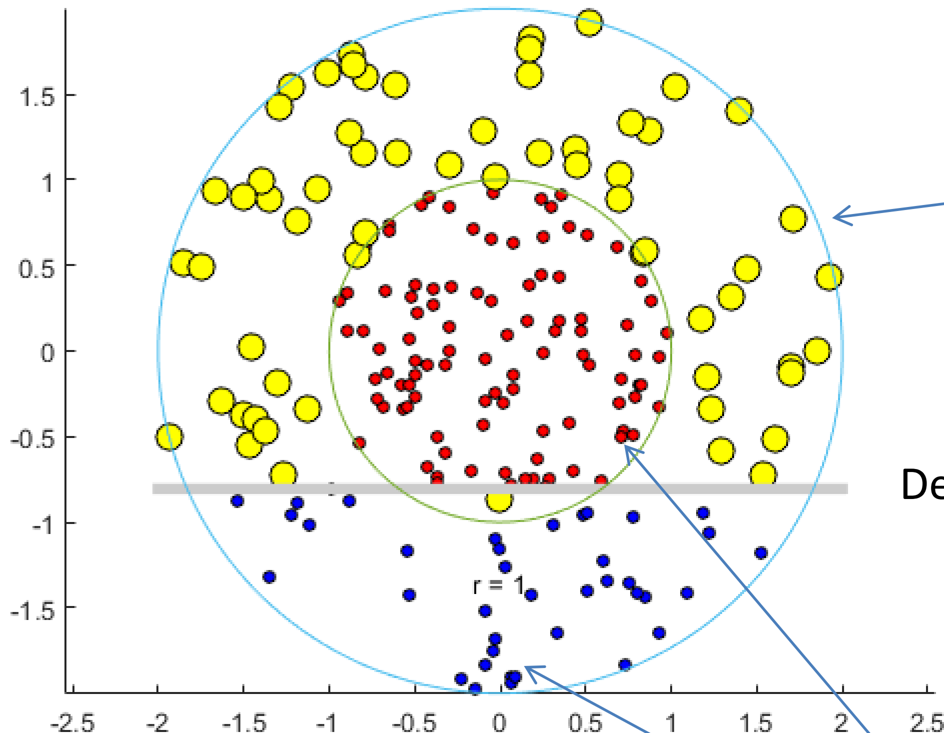
Boosting Example: the Dataset



Setting:
-- Two-class problem
-- Inner ring: red class
-- Outer ring: blue class

-- Linear model as the base (weak) classifier
-- #weak classifier = 5

Iteration 1



The yellow points are incorrectly classified. They get higher weight for next iteration.

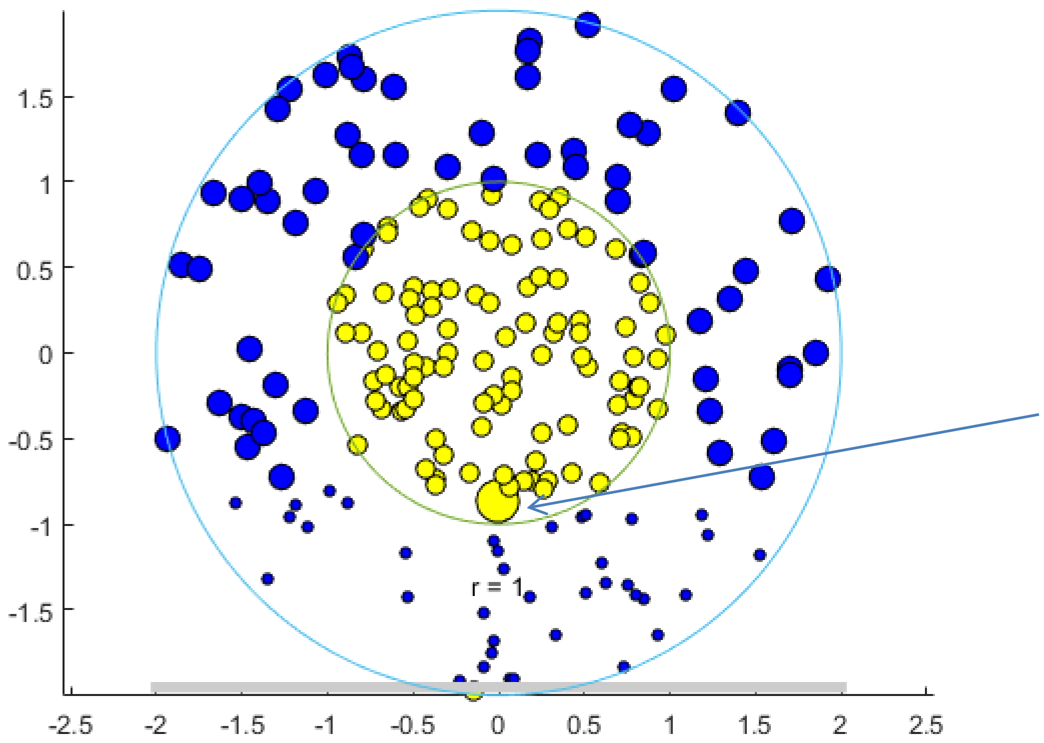
Red

Decision boundary

Blue

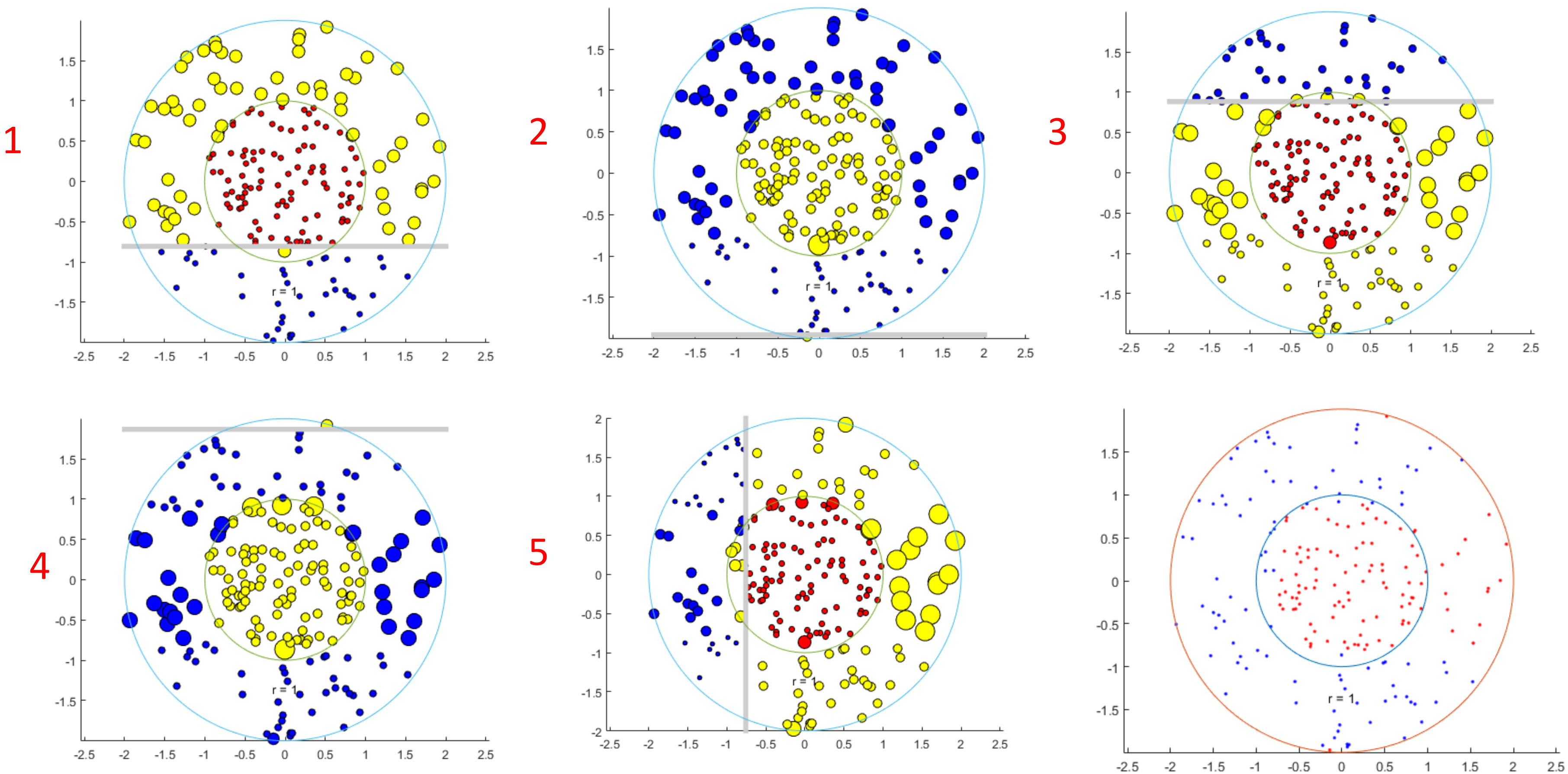
These points are correctly classified. They get lower weight for next iteration.

Iteration 2



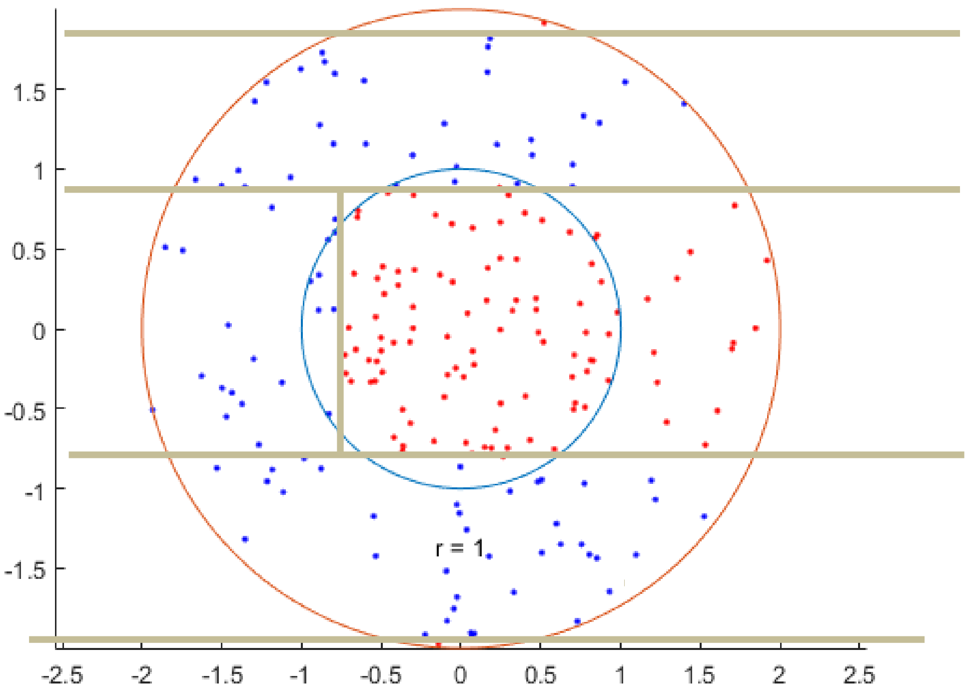
Incorrectly classified again!
This point gets even higher weight.

Together



The final result

The Combined Model

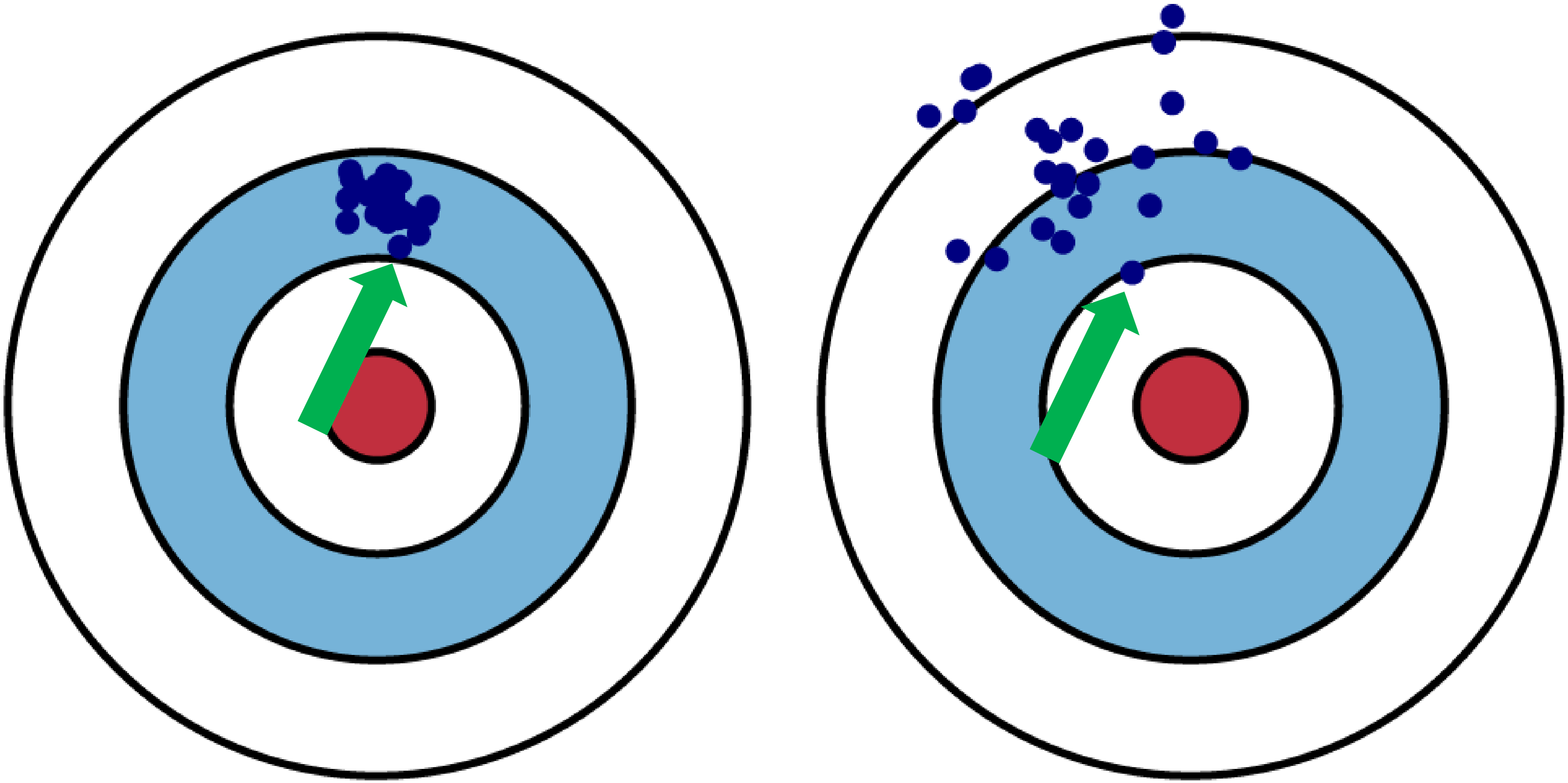


The (roughly) combined decision boundary. Much better than a single linear model!!!

Please input a larger `weakLearnerNum`

And your decision boundary will get closer and closer to the inner cycle.

Effect on Bias



Strong classifiers get higher class weight → push to the target function

Effect on Variance

- Not theoretically clear.
- In practice, boosting is more prone to overfitting.
- BUT, some recent studies claim that their boosting methods can reduce both bias and variance.

Resampling vs Reweighting

- Reweighting usually works better
- Resampling is easier to implement
- Reweighting is more sensitive to noise data
- Resampling doesn't work well on stable classifiers

Stacking

- Combine different TYPEs of classifiers
- Difficult to analyze theoretically
  * Just try and see how it goes….
- Less widely used due to the poor interpretability