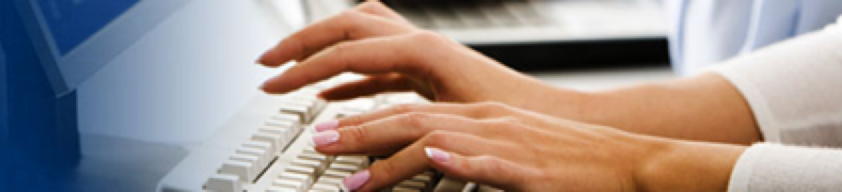


# Lecture 10: Ensemble methods

Meelis Kull

meelis.kull@ut.ee

Fall 2019

# Previous Lecture 09 – Deep learning

- ✓ Minimisation of average loss
- ✓ Compositional models
- ✓ Feed-forward neural networks
- ✓ Example network
- ✓ Back-propagation algorithm
- ✓ Universal approximation theorem
- ✓ Key challenges in deep learning
  - ✓ Choosing the structure
  - ✓ Regularisation
  - ✓ Optimisation
- ✓ Multi-class classification and softmax
- ✓ Convolutional neural networks and summary

# Lecture 10 – Ensemble methods

- Why do we need ensemble methods?
- Bagging
- Random forest
- Weighted averaging
- Boosting
  - Intuitive explanation
  - AdaBoost algorithm
  - Alternative formulations
  - Interpretations

# Acknowledgements


- This lecture is partly inspired by:
- Gavin Brown, University of Manchester
  - INIT/AERFAI Summer School on Machine Learning 3 lectures on Multiple Classifier Systems (2013)
- Raivo Kolde, University of Tartu
  - Machine Learning course, 1 lecture on Basics of Ensemble Methods (2012)
- Mari-Liis Allikivi and Ardi Tampuu, University of Tartu
  - Seminar presentation on boosting within the Special Course in Machine Learning: Ensemble methods (2017)
- Some slides have been reused as indicated on the slides

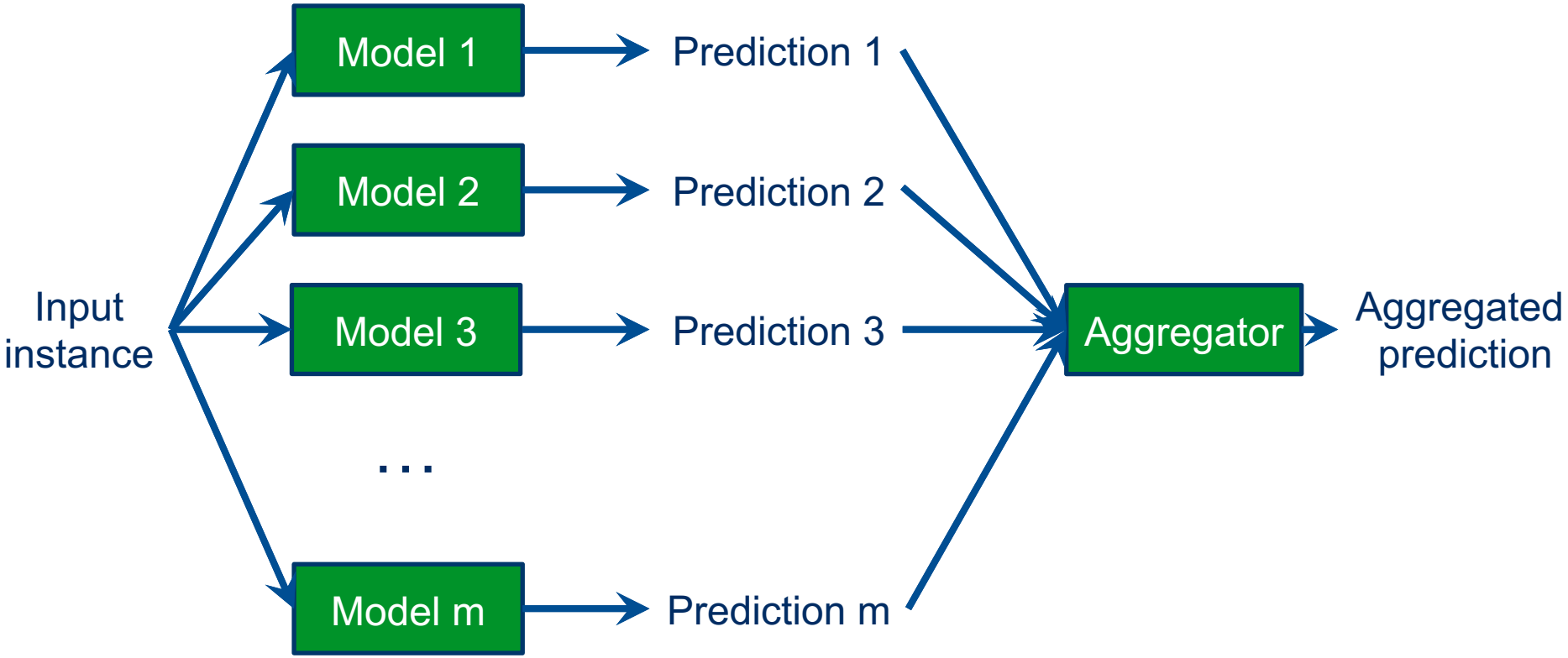
# Lecture 10 – Ensemble methods

- **Why do we need ensemble methods?**
- Bagging
- Random forest
- Weighted averaging
- Boosting
  - Intuitive explanation
  - AdaBoost algorithm
  - Alternative formulations
  - Interpretations

# What are Ensemble Methods?

- Ensemble method:
  - Get predictions from multiple models (*ensemble*)
    and aggregate the predictions

# Aggregation of predictions



# What are Ensemble Methods?

- Ensemble method:
  - Get predictions from multiple models (*ensemble*)
    and aggregate the predictions
- Key questions in creating an ensemble method:
  - How to get multiple models?
  - How to aggregate the predictions?

# Example of an ensemble method

- County fair in Cornwall, England in 1906
- Competition: Guess the weight of the cow
- 787 participants
- Correct answer:
  - 1198 lb ~ 543 kg
- Sir Francis Galton recorded the results and published in Nature

mean of the ire for month

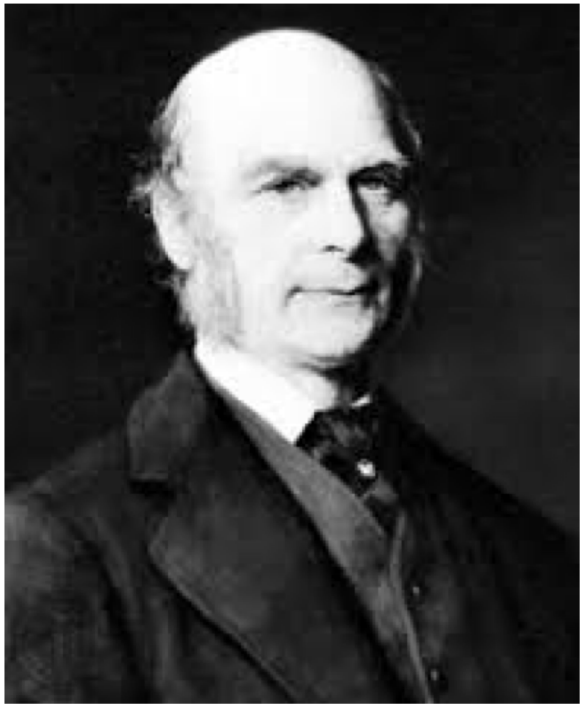
l year- Both years. Bulletin ontains station ults of ions in 51 and y in a ation is and is l appli- o with mpera- aximum bsolute bsolute l rain- nt was . Most tember. it being J. D.

results leutsche second

*Distribution of the estimates of the dressed weight of a particular living ox, made by 787 different persons.*

| Degrees of the length of Array 0°—100° | Estimates in lbs. | Centiles: Observed deviates from 1207 lbs. | Centiles: Normal p.e =37 | Excess of Observed over Normal |
|---|---|---|---|---|
| 5 | 1074 | −133 | −90 | +43 |
| 10 | 1109 | − 98 | −70 | +28 |
| 15 | 1126 | − 81 | −57 | +24 |
| 20 | 1148 | − 59 | −46 | +13 |
| $q_1$ 25 | 1162 | − 45 | −37 | + 8 |
| 30 | 1174 | − 33 | −29 | + 4 |
| 35 | 1181 | − 26 | −21 | + 5 |
| 40 | 1188 | − 19 | −14 | + 5 |
| 45 | 1197 | − 10 | − 7 | + 3 |
| $m$ 50 | 1207 | 0 | 0 | 0 |
| 55 | 1214 | + 7 | + 7 | 0 |
| 60 | 1219 | + 12 | +14 | − 2 |
| 65 | 1225 | + 18 | +21 | − 3 |
| 70 | 1230 | + 23 | +29 | − 6 |
| $q_3$ 75 | 1236 | + 29 | +37 | − 8 |
| 80 | 1243 | + 36 | +46 | −10 |
| 85 | 1254 | + 47 | +57 | −10 |
| 90 | 1267 | + 52 | +70 | −18 |
| 95 | 1293 | + 86 | +90 | − 4 |

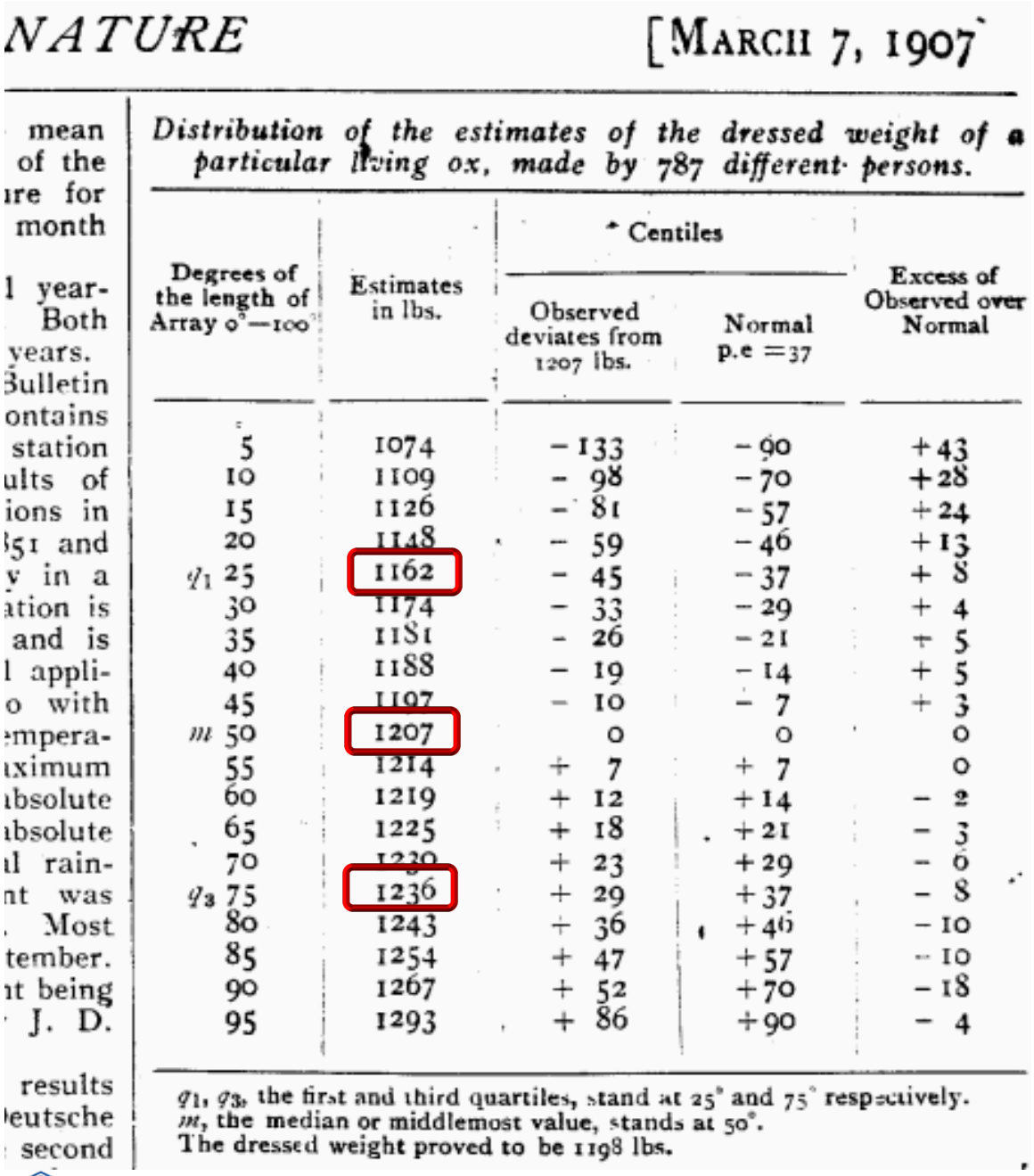
$q_1$, $q_3$, the first and third quartiles, stand at 25° and 75° respectively.
$m$, the median or middlemost value, stands at 50°.
The dressed weight proved to be 1198 lbs.



Sir Francis Galton

Francis Galton
VOX POPULI
Nature (1907),
No. 1949, Vol. 75, 450-451

http://wisdomofcrowds.blogspot.co.uk/2009/12/vox-populi-sir-francis-galton.html



NATURE [MARCH 7, 1907

*Distribution of the estimates of the dressed weight of a particular living ox, made by 787 different persons.*

| Degrees of the length of Array 0°—100° | Estimates in lbs. | Centiles: Observed deviates from 1207 lbs. | Centiles: Normal p.e = 37 | Excess of Observed over Normal |
|---|---|---|---|---|
| 5 | 1074 | −133 | −90 | +43 |
| 10 | 1109 | − 98 | −70 | +28 |
| 15 | 1126 | − 81 | −57 | +24 |
| 20 | 1148 | − 59 | −46 | +13 |
| $q_1$ 25 | 1162 | − 45 | −37 | + 8 |
| 30 | 1174 | − 33 | −29 | + 4 |
| 35 | 1181 | − 26 | −21 | + 5 |
| 40 | 1188 | − 19 | −14 | + 5 |
| 45 | 1197 | − 10 | − 7 | + 3 |
| $m$ 50 | 1207 | 0 | 0 | 0 |
| 55 | 1214 | + 7 | + 7 | 0 |
| 60 | 1219 | + 12 | +14 | − 2 |
| 65 | 1225 | + 18 | +21 | − 3 |
| 70 | 1230 | + 23 | +29 | − 6 |
| $q_3$ 75 | 1236 | + 29 | +37 | − 8 |
| 80 | 1243 | + 36 | +46 | −10 |
| 85 | 1254 | + 47 | +57 | −10 |
| 90 | 1267 | + 52 | +70 | −18 |
| 95 | 1293 | + 86 | +90 | − 4 |

$q_1$, $q_3$, the first and third quartiles, stand at 25° and 75° respectively.
$m$, the median or middlemost value, stands at 50°.
The dressed weight proved to be 1198 lbs.

- Truth: 1198
- Percentiles:
  - 25th : 1162
  - 50th : 1207
  - 75th : 1236

mean
of the
ire for
month

l year-
Both
years.
Bulletin
ontains
station
ults of
ions in
51 and
y in a
ation is
and is
l appli-
o with
empera-
aximum
bsolute
bsolute
l rain-
nt was
. Most
tember.
it being
J. D.

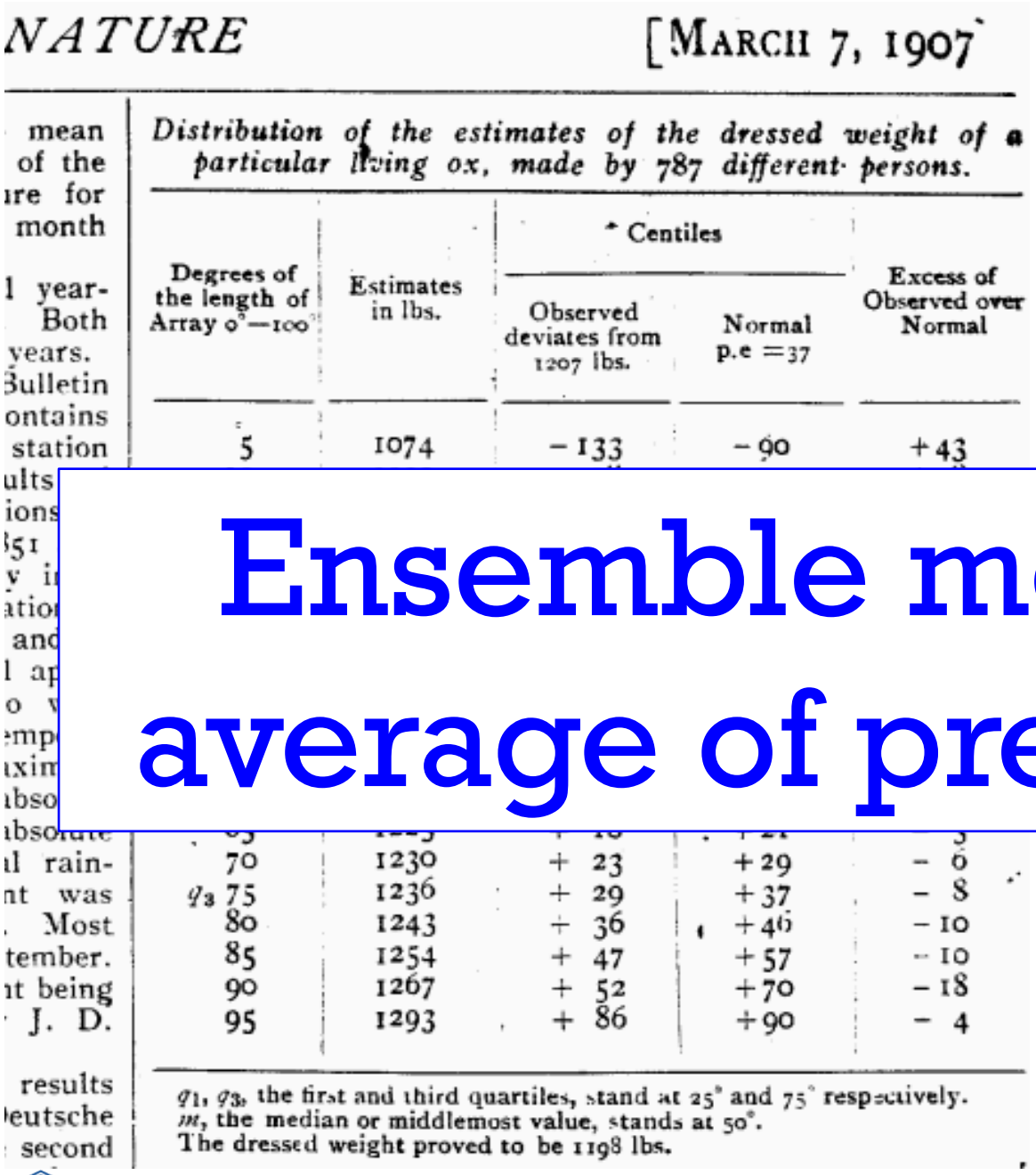
results
Deutsche
second

*Distribution of the estimates of the dressed weight of a particular living ox, made by 787 different persons.*

| Degrees of the length of Array 0°—100° | Estimates in lbs. | Centiles: Observed deviates from 1207 lbs. | Centiles: Normal p.e = 37 | Excess of Observed over Normal |
|---|---|---|---|---|
| 5 | 1074 | −133 | −90 | +43 |
| 10 | 1109 | − 98 | −70 | +28 |
| 15 | 1126 | − 81 | −57 | +24 |
| 20 | 1148 | − 59 | −46 | +13 |
| $q_1$ 25 | 1162 | − 45 | −37 | + 8 |
| 30 | 1174 | − 33 | −29 | + 4 |
| 35 | 1181 | − 26 | −21 | + 5 |
| 40 | 1188 | − 19 | −14 | + 5 |
| 45 | 1197 | − 10 | − 7 | + 3 |
| $m$ 50 | 1207 | 0 | 0 | 0 |
| 55 | 1214 | + 7 | + 7 | 0 |
| 60 | 1219 | + 12 | +14 | − 2 |
| 65 | 1225 | + 18 | +21 | − 3 |
| 70 | 1230 | + 23 | +29 | − 6 |
| $q_3$ 75 | 1236 | + 29 | +37 | − 8 |
| 80 | 1243 | + 36 | +46 | −10 |
| 85 | 1254 | + 47 | +57 | −10 |
| 90 | 1267 | + 52 | +70 | −18 |
| 95 | 1293 | + 86 | +90 | − 4 |

$q_1$, $q_3$, the first and third quartiles, stand at 25° and 75° respectively.
$m$, the median or middlemost value, stands at 50°.
The dressed weight proved to be 1198 lbs.

- Truth: 1198
- Percentiles:
  - $25^{th}$ : 1162
  - $50^{th}$ : 1207
  - $75^{th}$ : 1236
- Mean: 1197 !!!!!



NATURE [MARCH 7, 1907

*Distribution of the estimates of the dressed weight of a particular living ox, made by 787 different persons.*

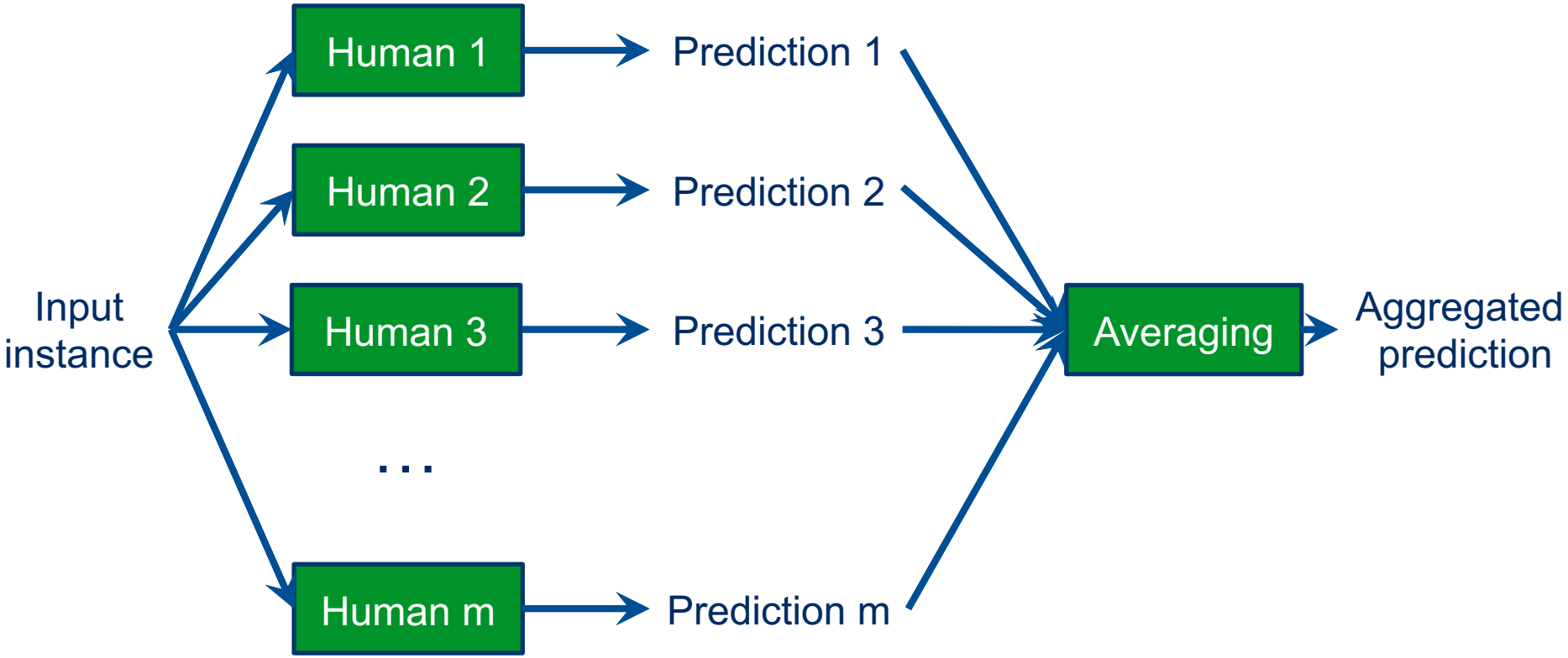
| Degrees of the length of Array 0°—100° | Estimates in lbs. | Centiles: Observed deviates from 1207 lbs. | Centiles: Normal p.e = 37 | Excess of Observed over Normal |
|---|---|---|---|---|
| 5 | 1074 | − 133 | − 90 | + 43 |
| 70 | 1230 | + 23 | + 29 | − 6 |
| $q_3$ 75 | 1236 | + 29 | + 37 | − 8 |
| 80 | 1243 | + 36 | + 46 | − 10 |
| 85 | 1254 | + 47 | + 57 | − 10 |
| 90 | 1267 | + 52 | + 70 | − 18 |
| 95 | 1293 | + 86 | + 90 | − 4 |

$q_1$, $q_3$, the first and third quartiles, stand at 25° and 75° respectively.
$m$, the median or middlemost value, stands at 50°.
The dressed weight proved to be 1198 lbs.

# Ensemble method: average of predictors

- Truth: 1198
- Percentiles: 2 7 6
- Mean: 1197 !!!!!

# Aggregation of predictions



# A theoretical justification to averaging

- $t$ - truth; $x_1, x_2, \ldots, x_M$ - predictions
- $x_{rand}$ - randomly selecting one of these

$$\epsilon_{rand} = \mathbb{E}[(x_{rand} - t)^2] = \frac{1}{M}\sum_{i=1}^{M}(x_i - t)^2$$

- $\overline{x} = \frac{1}{M}\sum_{i=1}^{M} x_i$ - average prediction

$$\epsilon_{aver} = (\overline{x} - t)^2$$

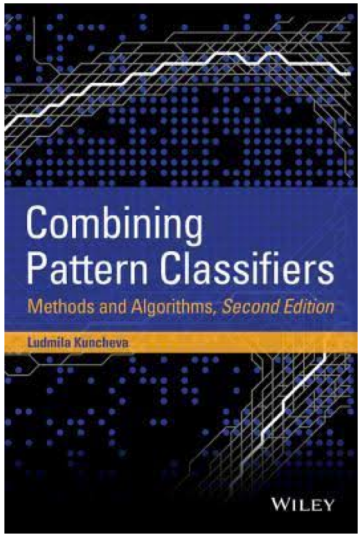
- Well-known "ambiguity" decompositon:

$$\epsilon_{rand} = \epsilon_{aver} + \frac{1}{M}\sum_{i=1}^{M}(x_i - \overline{x})^2$$

$$\Longrightarrow \quad \epsilon_{aver} \leq \epsilon_{rand} \qquad !!!!!$$

# Reasons to use ensembles

- **Reasons** (From the 2004 book by Ludmila Kuncheva *Combining pattern classifiers: methods and algorithms*)
  - Statistical:
    - Hopefully the ensemble generalises better than a single chosen model
  - Computational
    - Averaging can sometimes be a fast way of reaching closer to the optimal than direct optimisation
  - Representational
    - Averaging models of some model class can sometimes take you outside of that model class



# In which supervised learning tasks can ensemble models be used?

A. Only classification

B. Only regression

C. Classification and regression

✓ D. All supervised learning tasks

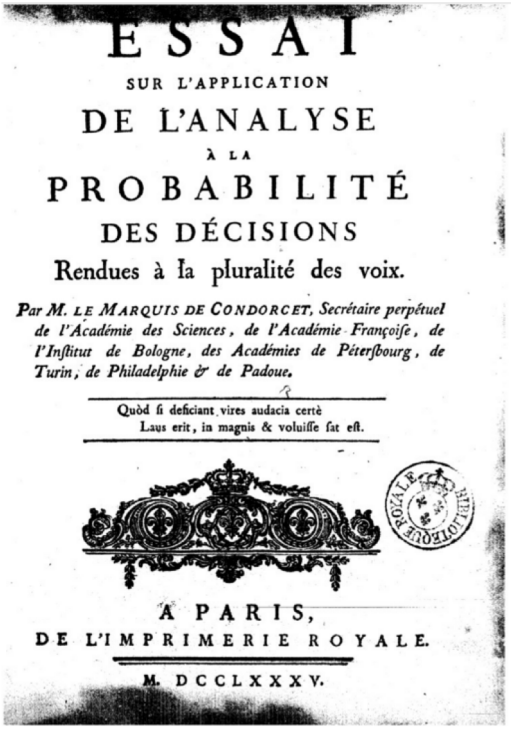
E. I don't know

# Types of tasks in supervised learning where ensembles can be used

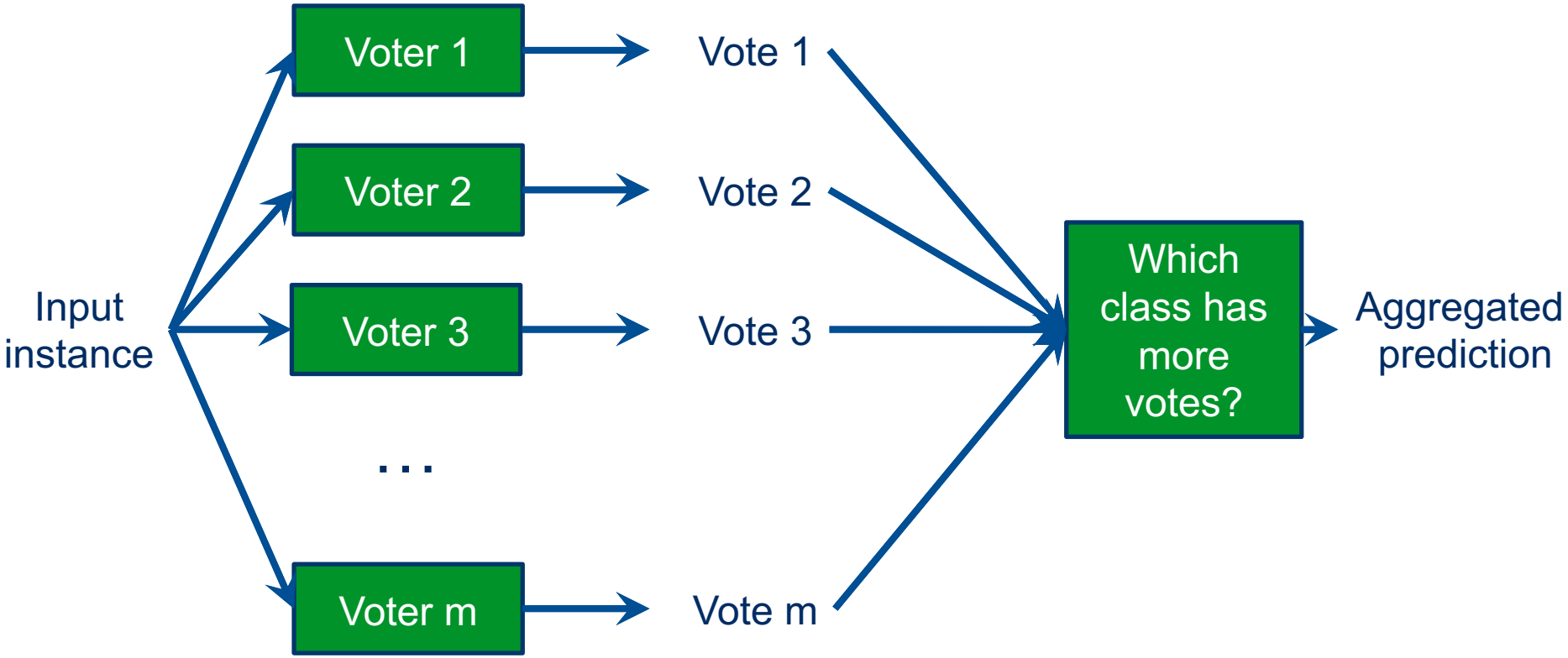
- Ensembles can be used in pretty much all supervised learning tasks
  - Classification
  - Regression
  - Structured output prediction

# Theory in favour of voting (binary task)

- Marquis de Condorcet 1785:
  Essay on the Application of Analysis to the Probability of Majority Decisions

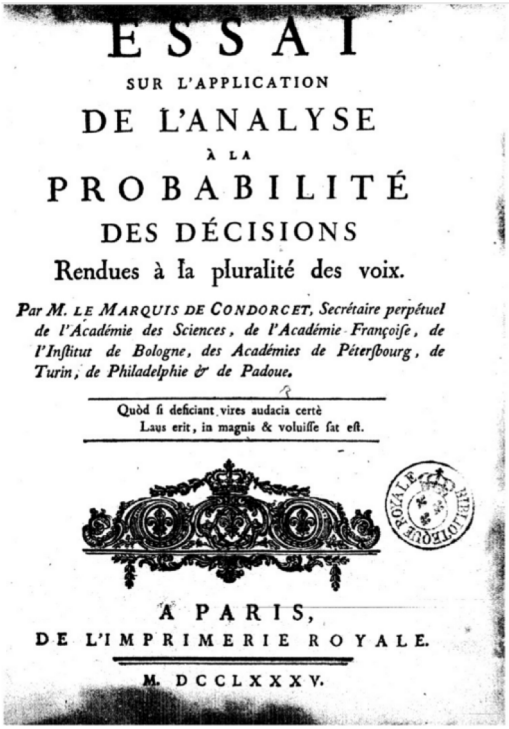

ESSAI
SUR L'APPLICATION
DE L'ANALYSE
À LA
PROBABILITÉ
DES DÉCISIONS
Rendues à la pluralité des voix.
Par M. LE MARQUIS DE CONDORCET, Secrétaire perpétuel de l'Académie des Sciences, de l'Académie Françoise, de l'Institut de Bologne, des Académies de Pétersbourg, de Turin, de Philadelphie & de Padoue.
Quòd si deficiant vires audacia certè
Laus erit, in magnis & voluisse sat est.
A PARIS,
DE L'IMPRIMERIE ROYALE.
M. DCCLXXXV.

# Voting in binary classification



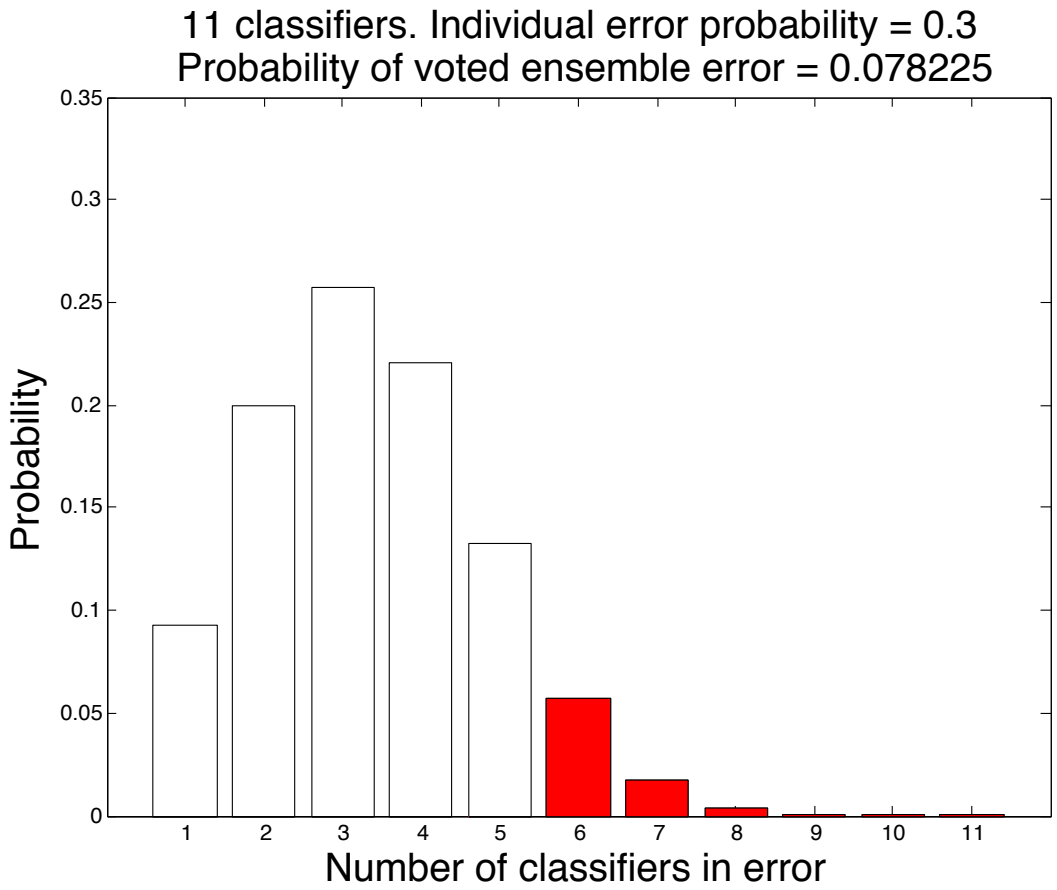
# Theory in favour of voting (binary task)

- Marquis de Condorcet 1785:
  Essay on the Application of Analysis to the Probability of Majority Decisions
- $M$ voters, independent errors, individual error probability $\epsilon$
- Majority vote is wrong with probability:


ESSAI
SUR L'APPLICATION
DE L'ANALYSE
À LA
PROBABILITÉ
DES DÉCISIONS
Rendues à la pluralité des voix.
Par M. LE MARQUIS DE CONDORCET, Secrétaire perpétuel de l'Académie des Sciences, de l'Académie Françoise, de l'Institut de Bologne, des Académies de Pétersbourg, de Turin, de Philadelphie & de Padoue.
Quòd si deficiant vires audacia certè
Laus erit, in magnis & voluisse sat est.
A PARIS,
DE L'IMPRIMERIE ROYALE.
M. DCCLXXXV.

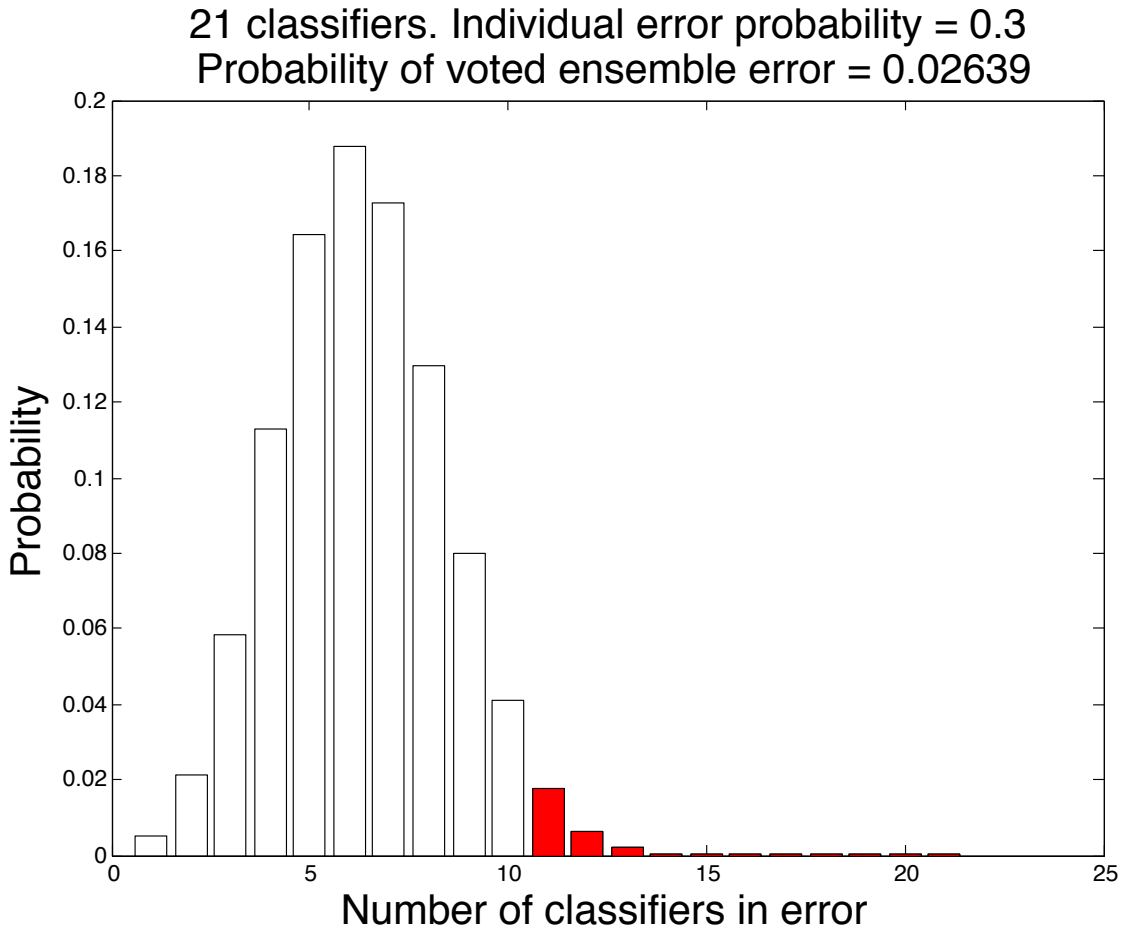
$$\sum_{k \geq \lceil \frac{M+1}{2} \rceil} \binom{M}{k} \epsilon^k (1-\epsilon)^{M-k}$$

$$p(majority\ vote\ error) = \sum_{k \geq \lceil \frac{M+1}{2} \rceil} \binom{M}{k} \epsilon^k (1-\epsilon)^{(M-k)}$$



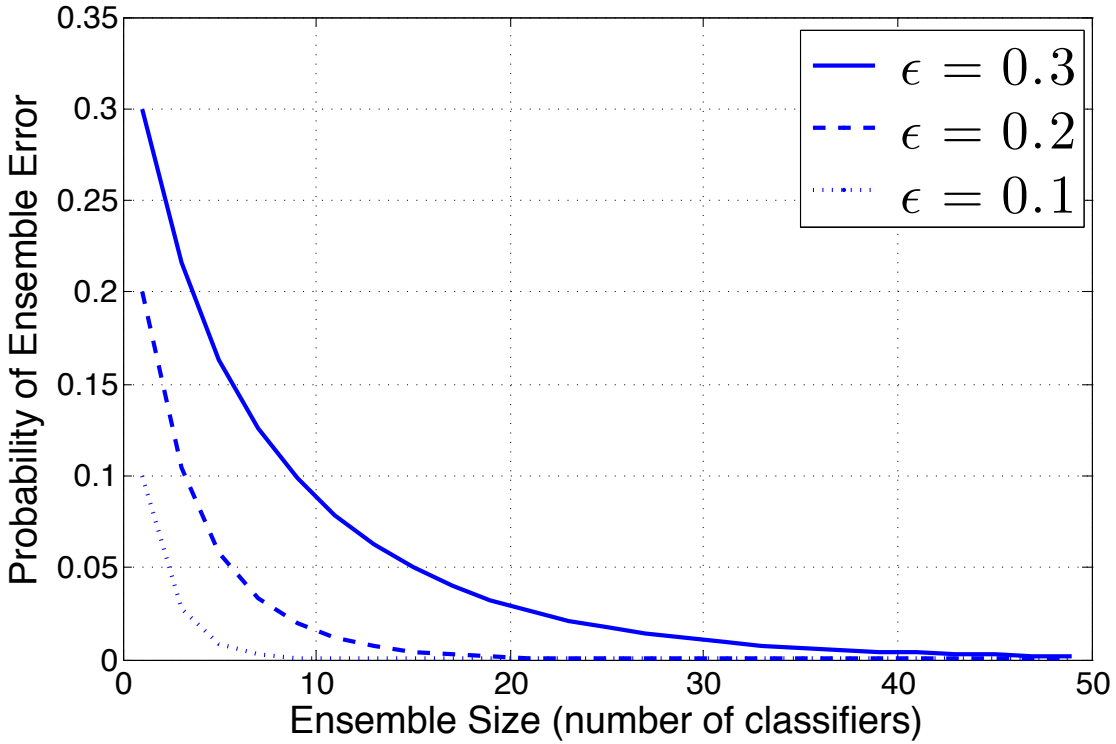

Slide adapted from Gavin Brown

$$p(majority\ vote\ error) = \sum_{k \geq \lceil \frac{M+1}{2} \rceil} \binom{M}{k} \epsilon^k (1-\epsilon)^{(M-k)}$$




Slide adapted from Gavin Brown

$$p(majority\ vote\ error) = \sum_{k \geq \lceil \frac{M+1}{2} \rceil} \binom{M}{k} \epsilon^k (1-\epsilon)^{(M-k)}$$




Virtually ZERO error by M = 50 !!

**Slide adapted from Gavin Brown**

# Why does this theory not work in practice?

# Why does this theory not work in practice?

A. Too few training data
B. Individual models are not good enough
✓C. Errors of individual models are not independent
D. Not enough different learning algorithms exist
E. I don't know

# Why does this theory not work in practice?

Errors of voters are not independent

# Key challenge of ensemble learning

- The key challenge of ensemble learning is to obtain models that are:
  - reasonably accurate
  - as independent as possible

# Lecture 10 – Ensemble methods

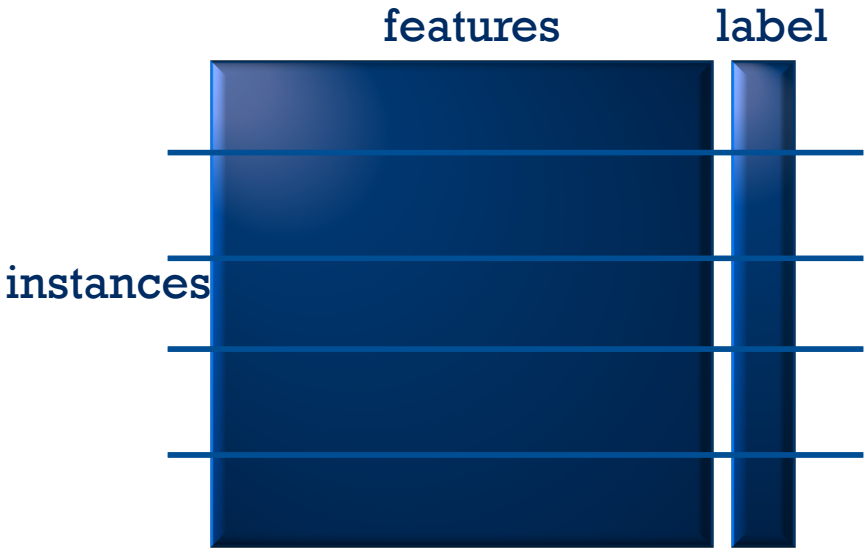
- ✓ Why do we need ensemble methods?
- **Bagging**
- Random forest
- Weighted averaging
- Boosting
  - Intuitive explanation
  - AdaBoost algorithm
  - Alternative formulations
  - Interpretations

# How to achieve independence of models?

- Idea 1: Split+Train
  - Randomly split training data into M disjoint groups of instances, train a separate model on each group

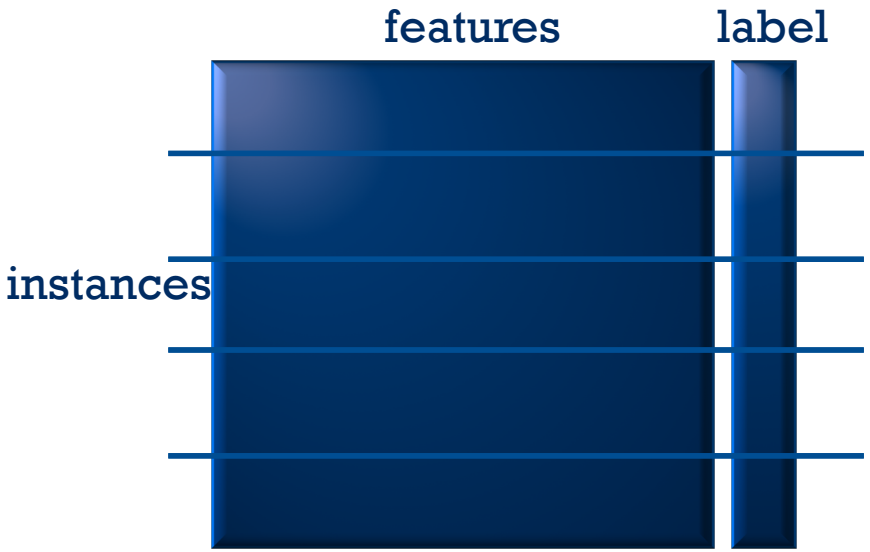
# How to achieve independence of models?

- Idea 1: Split+Train
  - Randomly split training data into M disjoint groups of instances, train a separate model on each group



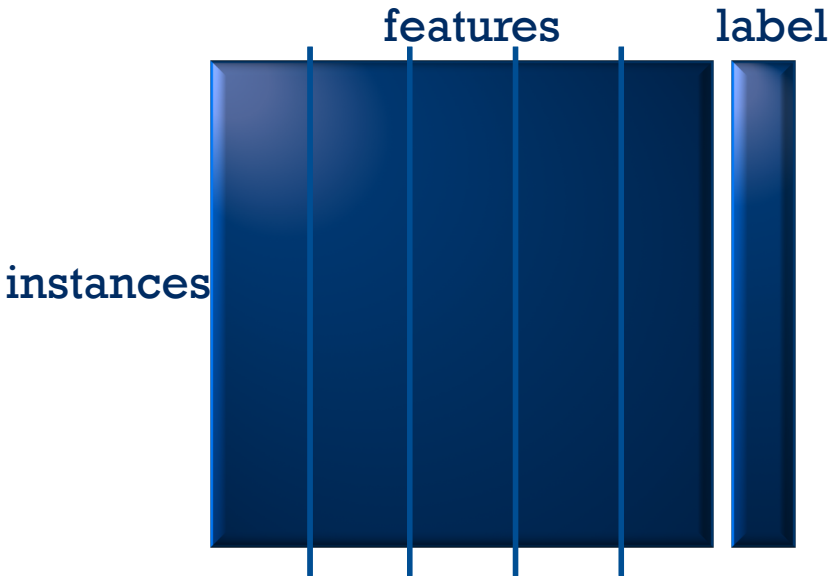
# How to achieve independence of models?

- Idea 1: Split+Train
  - Randomly split training data into M disjoint groups of instances, train a separate model on each group
  - Bad because groups get small and the learned models will have poor prediction quality



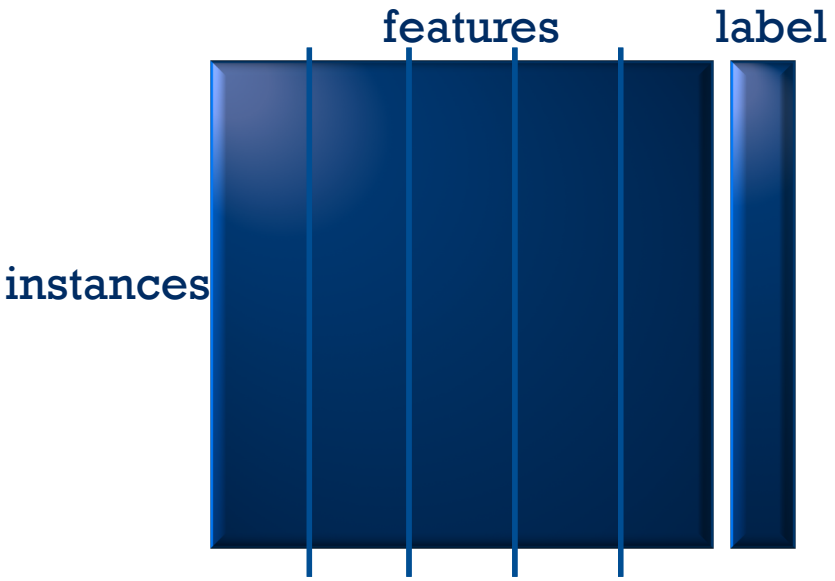
# How to achieve independence of models?

- Idea 2: Split by features+Train
  - Randomly split training data into M disjoint groups of features, train a separate model on each group



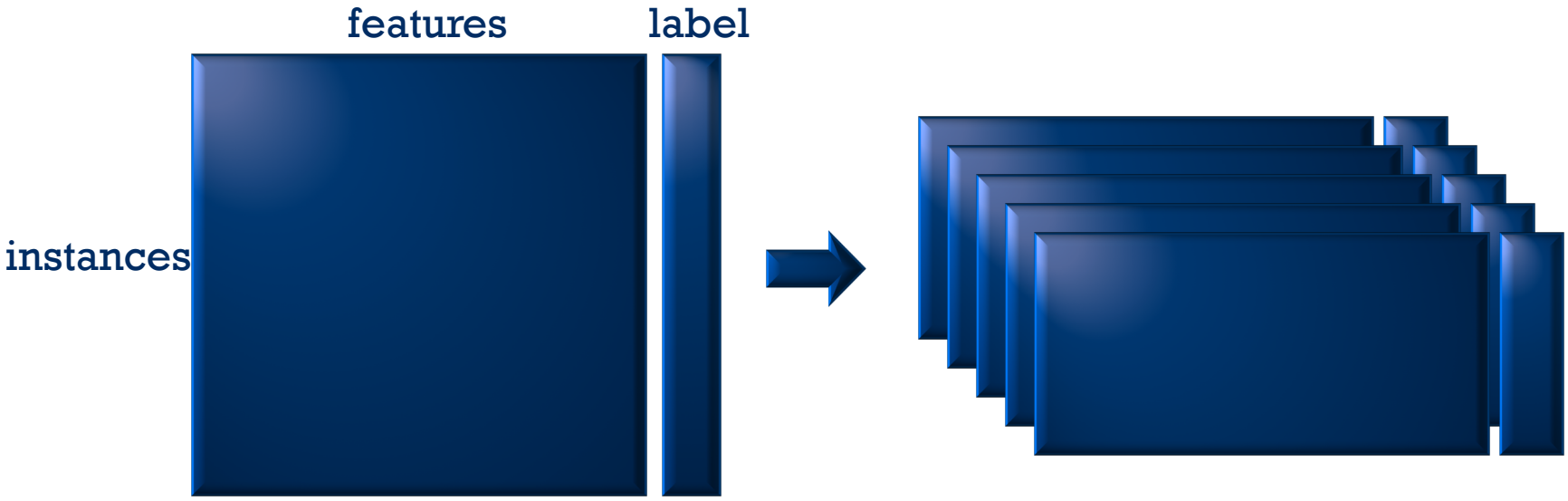
# How to achieve independence of models?

- Idea 2: Split by features+Train
  - Randomly split training data into M disjoint groups of features, train a separate model on each group
  - Lack of good features can be even worse than lack of instances



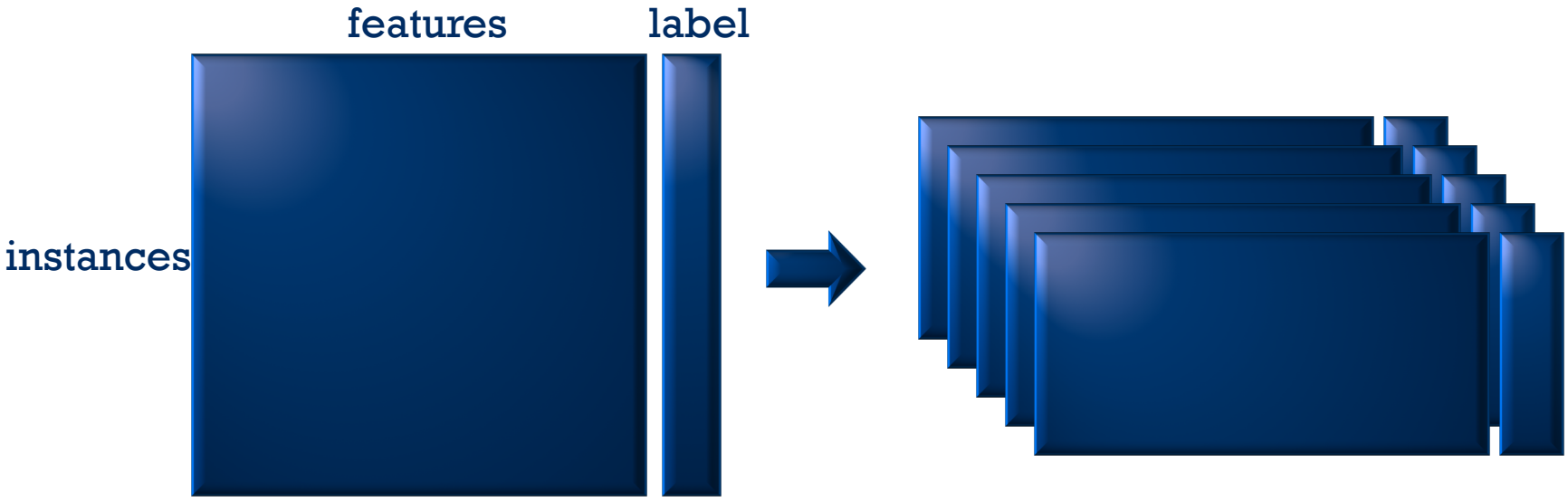
# How to achieve independence of models?

- Idea 3: Overlapping subsets of instances+Train
  - Randomly sample M overlapping groups of instances, train a separate model on each group



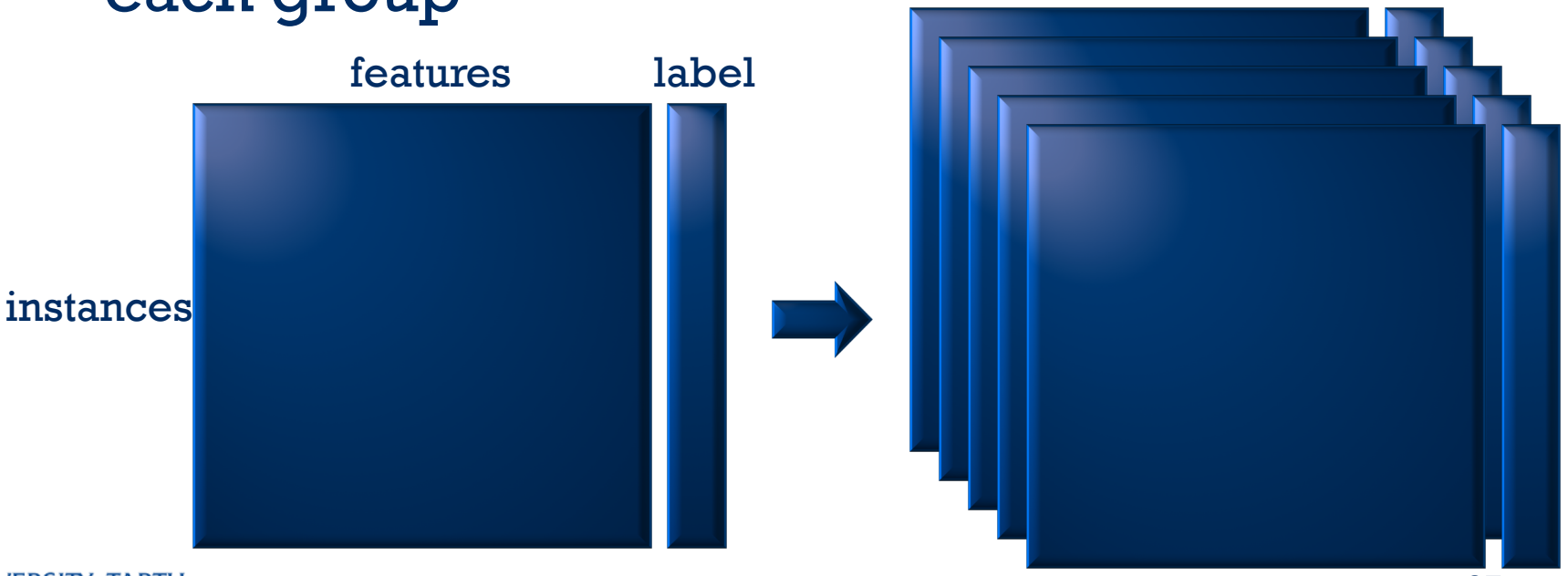
# How to achieve independence of models?

- Idea 3: Overlapping subsets of instances+Train
  - Randomly sample M overlapping groups of instances, train a separate model on each group
  - Not too bad, but still smaller training sets



# How to achieve independence of models?

- Idea 4: Bootstrap-sampled instances+Train
  - Sample with replacement M overlapping groups of instances with same size as original (bootstrapping), train a separate model on each group



# Generating a new dataset by "Bootstrapping"

- Bootstrapping:
  - Sample N items with replacement from the original N training instances

**Original data**

| id | x1 | x2 | x3 | y |
|---|---|---|---|---|
| 1 | 0.18 | 0.45 | 0.8 | 0 |
| 2 | 0.11 | 0.82 | 0.07 | 0 |
| 3 | 0.87 | 0.3 | 0.21 | 1 |
| 4 | 0.34 | 0.49 | 0.18 | 1 |
| 5 | 0.95 | 0.64 | 0.63 | 0 |
| 6 | 0.03 | 0.59 | 0.15 | 1 |

**Bootstrap 1**

| id | x1 | x2 | x3 | y |
|---|---|---|---|---|
| 1 | 0.18 | 0.45 | 0.8 | 0 |
| 1 | 0.18 | 0.45 | 0.8 | 0 |
| 3 | 0.87 | 0.3 | 0.21 | 1 |
| 4 | 0.34 | 0.49 | 0.18 | 1 |
| 5 | 0.95 | 0.64 | 0.63 | 0 |
| 5 | 0.95 | 0.64 | 0.63 | 0 |

**Bootstrap 2**

| id | x1 | x2 | x3 | y |
|---|---|---|---|---|
| 1 | 0.18 | 0.45 | 0.8 | 0 |
| 2 | 0.11 | 0.82 | 0.07 | 0 |
| 3 | 0.87 | 0.3 | 0.21 | 1 |
| 6 | 0.03 | 0.59 | 0.15 | 1 |
| 6 | 0.03 | 0.59 | 0.15 | 1 |
| 6 | 0.03 | 0.59 | 0.15 | 1 |

Slide adapted from Gavin Brown

# Is it possible that a particular training instance is not included in the bootstrap sample?

A. Not possible

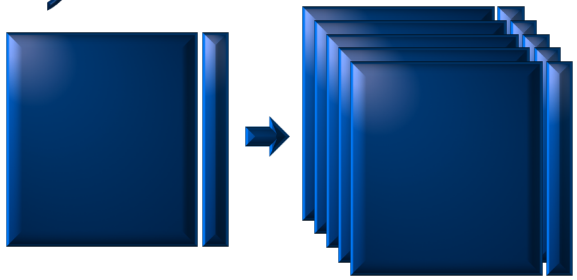
B. Possible, but only when the training set is small

✓C. Possible, happens with less than 50% probability

D. Possible, happens with more than 50% probability

E. I don't know

# Bagging: Bootstrap AGGregating (Breiman 1996)

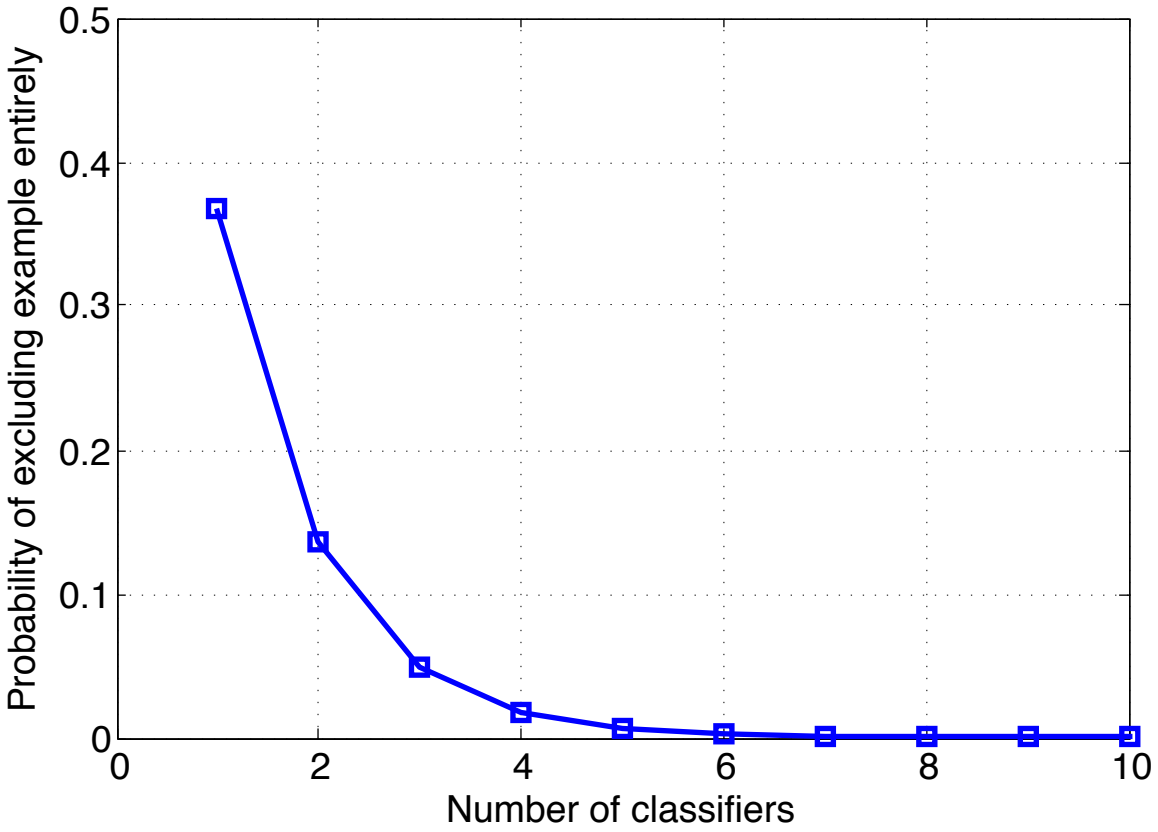


- Bootstrap sampling:
  - Sample with replacement M overlapping groups of instances with the same size as original
  - That is, each original instance will have 0, 1, or more copies in such a group
  - On average, each sample will have $1 - \frac{1}{e} \approx 63.2\%$ of original instances represented in the group
- Bagging (Bootstrap AGGregating = BAGG):
  - Train a model separately on each bootstrapped dataset and then aggregate the results

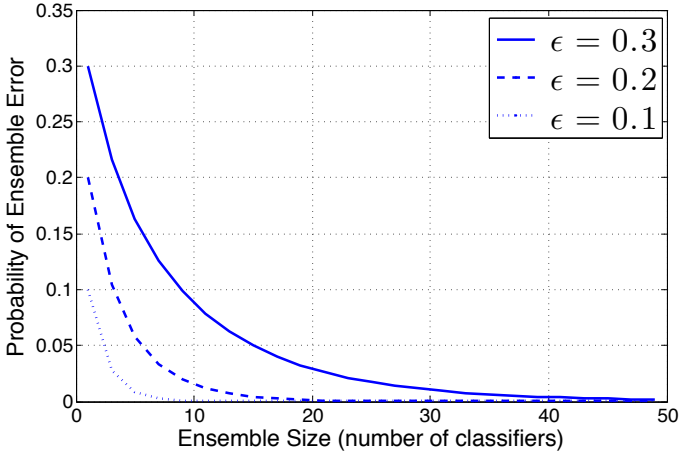
# Is it possible that a particular training instance is not included in any of the M bootstrapped datasets?

A. Not possible
B. Possible only when M is small
C. Possible only when the training set size N is small
✓D. Possible, but happens less and less frequently as M grows
E. Possible, but happens less and less frequently as N grows
F. I don't know

# Prob( *excluding an example from the whole ensemble* ) =

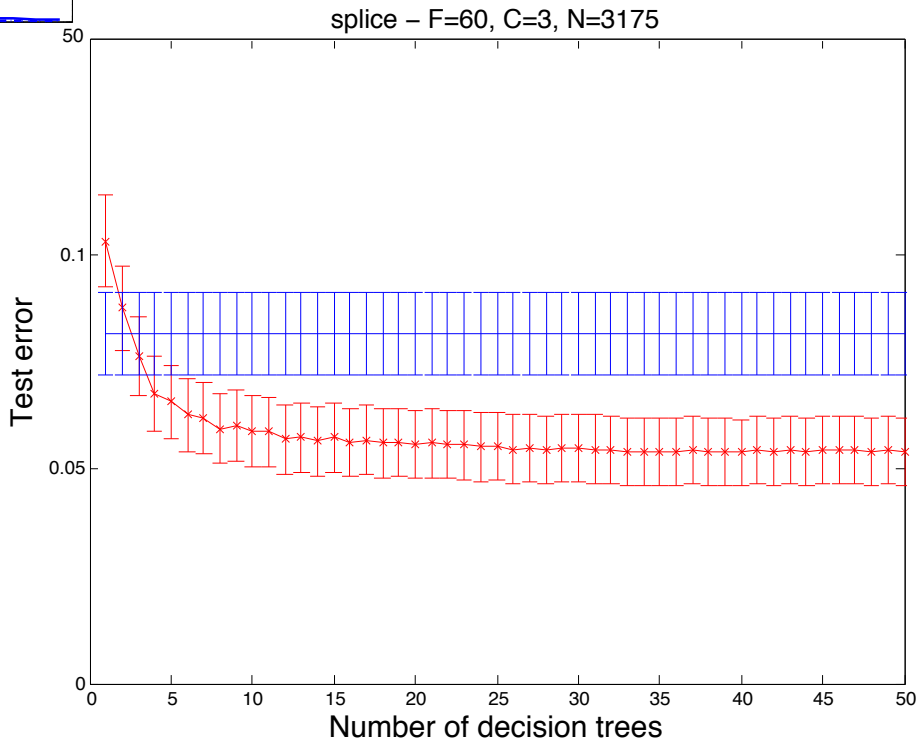



Slide adapted from Gavin Brown




Dream!

Reality!




Single decision tree

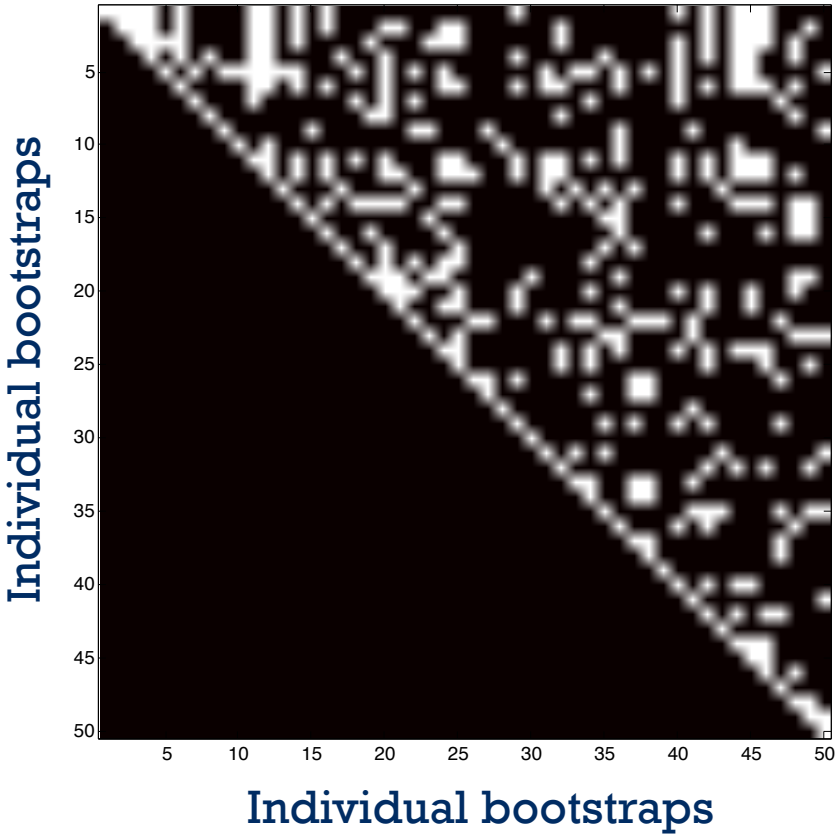
Bagging

Slide adapted from Gavin Brown

Independence test between errors of bagged decision trees.

$$(\chi^2 \text{ test}, \alpha = 0.05)$$

White pixel indicates significant correlation between errors of 2 individual decision trees in the ensemble




Slide adapted from Gavin Brown

# When is bagging useful?

- Bagging is bad if models are very similar (not independent enough)
- This happens if the learning algorithm is stable
  - That is, model does not usually change much after changing a few instances

## SVM?

# When is bagging useful?

- Bagging is bad if models are very similar (not independent enough)
- This happens if the learning algorithm is stable
  - That is, model does not usually change much after changing a few instances

# When is bagging useful?

- Bagging is bad if models are very similar (not independent enough)
- This happens if the learning algorithm is stable
  - That is, model does not usually change much after changing a few instances

# When is bagging useful?

- Bagging is bad if models are very similar (not independent enough)
- This happens if the learning algorithm is stable
  - That is, model does not usually change much after changing a few instances

# When is bagging useful?

- Bagging is bad if models are very similar (not independent enough)
- This happens if the learning algorithm is stable
  - That is, model does not usually change much after changing a few instances

# When is bagging useful?

- Bagging is bad if models are very similar (not independent enough)
- This happens if the learning algorithm is stable
  - That is, model does not usually change much after changing a few instances

# Bagging is strongly effected by the quality of individual models

Bagged decision stumps
(trees with 1 decision node)

Figure by Raivo Kolde, https://courses.cs.ut.ee/2012/ml/uploads/Main/lecture-18.pdf

# Summary of Bagging

- Individual models trained on bootstrap-sampled instances, predictions are aggregated
- Bagging is useful when the algorithm to learn individual models is:
  - Relatively accurate
  - Relatively unstable (high variance)
- The aggregated model is then usually better than the original model trained on full dataset

# Lecture 10 – Ensemble methods

- ✓ Why do we need ensemble methods?
- ✓ Bagging
- **Random forest**
- Weighted averaging
- Boosting
  - Intuitive explanation
  - AdaBoost algorithm
  - Alternative formulations
  - Interpretations

# Random forests (Breiman 2000):

- Random forests: similar to bagged decision trees but different in using features
- **In each recursive step** of learning decision trees:
  - **Randomly select F features** out of all P given features
  - Find the best split among these features
- Parameter F is usually fixed to be
  - F=sqrt(P) for classification
  - F=P/3 for regression

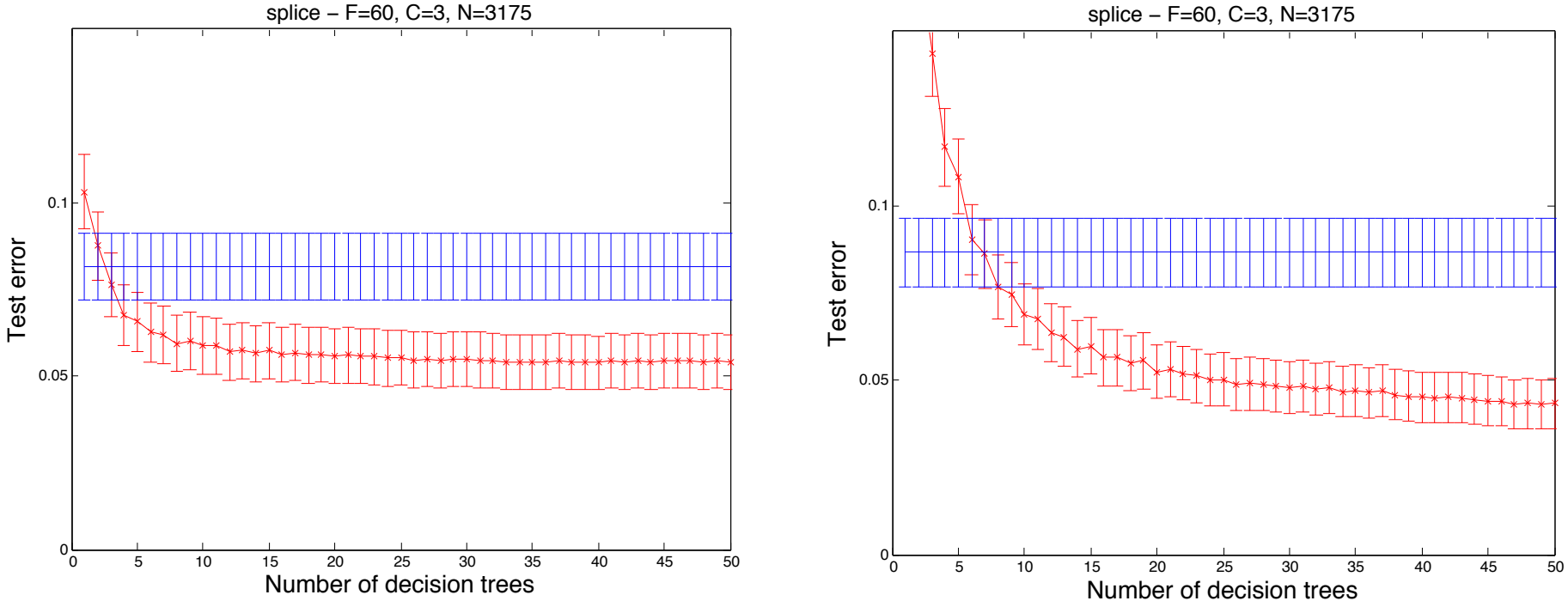



Figure 13: Bagging (LEFT) vs Random Forests (RIGHT) on the Splice dataset.

What do you notice about the starting point, for a single bagged/RF'd tree?

Slide adapted from Gavin Brown

***Real-time human pose recognition in parts from single depth images***
Computer Vision and Pattern Recognition 2011
Shotton et al, Microsoft Research


- *Basis of Kinect controller*
- *Features are simple image properties*
- *Test phase: 200 frames per sec on GPU*
- *Train phase more complex but still parallel*

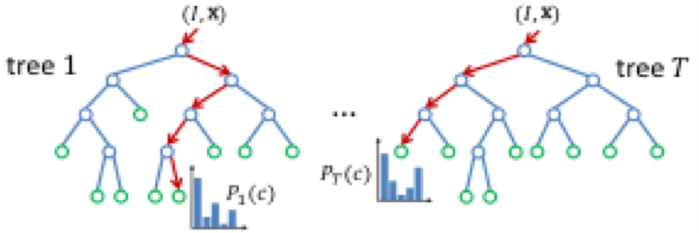



Figure 4. **Randomized Decision Forests.** A forest is an ensemble of trees. Each tree consists of split nodes (blue) and leaf nodes (green). The red arrows indicate the different paths that might be taken by different trees for a particular input.

### 3.3. Randomized decision forests

Randomized decision trees and forests [35, 30, 2, 8] have proven fast and effective multi-class classifiers for many tasks [20, 23, 36], and can be implemented efficiently on the GPU [34]. As illustrated in Fig. 4, a forest is an ensemble of $T$ decision trees, each consisting of split and leaf nodes.

To keep the training times down we employ a distributed implementation. Training 3 trees to depth 20 from 1 million images takes about a day on a 1000 core cluster.

Slide adapted from Gavin Brown

# How many trees in a forest?

# How many trees in a forest?

- The more the better!
- How do I know there are enough?
- Out-of-bag (OOB) error
  - For each training instance make a prediction using trees that do not use that instance and evaluate
  - Stabilisation of OOB error suggests that there are enough trees

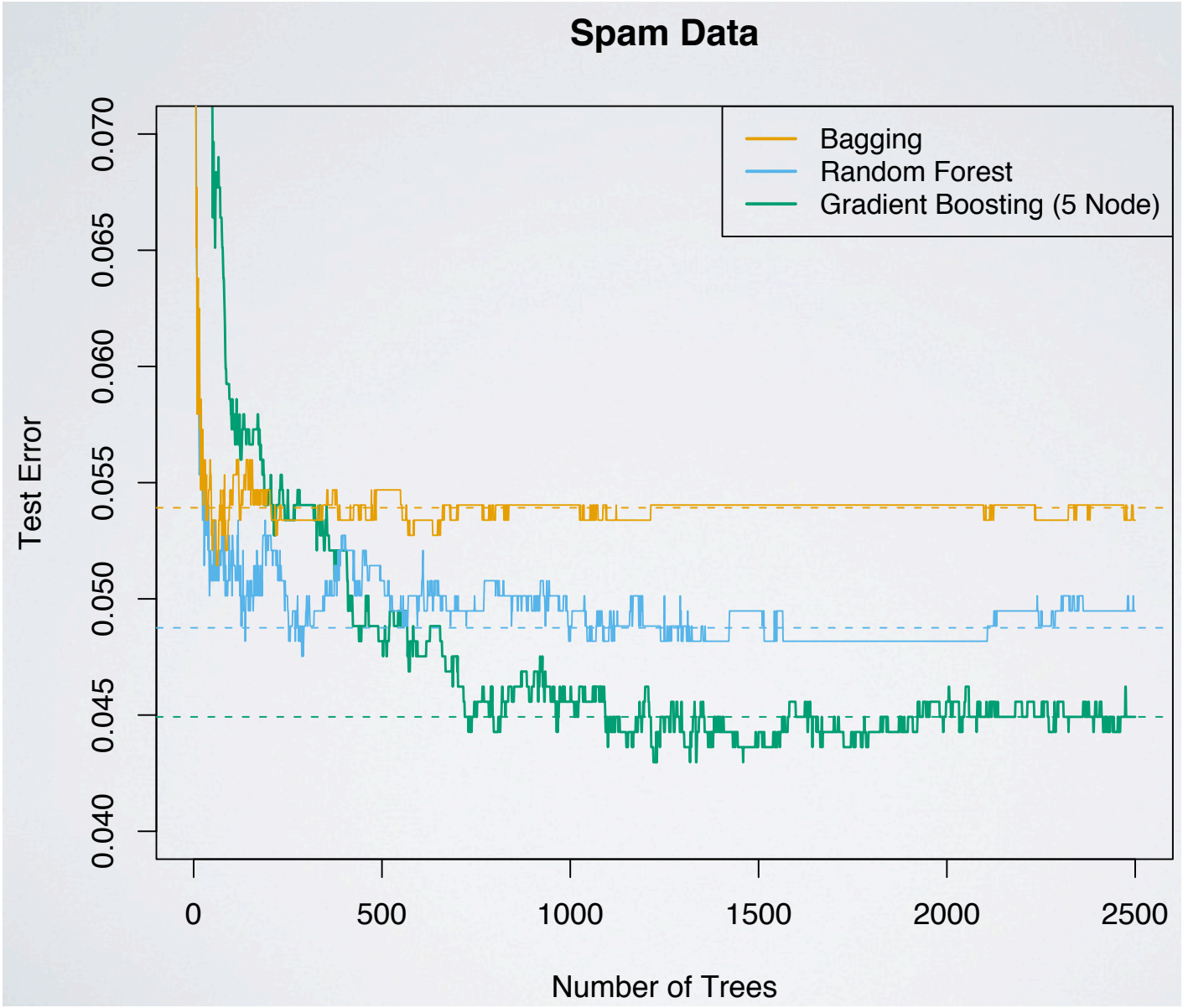



Figure by Raivo Kolde, https://courses.cs.ut.ee/2012/ml/uploads/Main/lecture-18.pdf

# Lecture 10 – Ensemble methods

- ✓ Why do we need ensemble methods?
- ✓ Bagging
- ✓ Random forest
- **Weighted averaging**
- Boosting
  - Intuitive explanation
  - AdaBoost algorithm
  - Alternative formulations
  - Interpretations

# Homogeneous and Heterogeneous ensembles

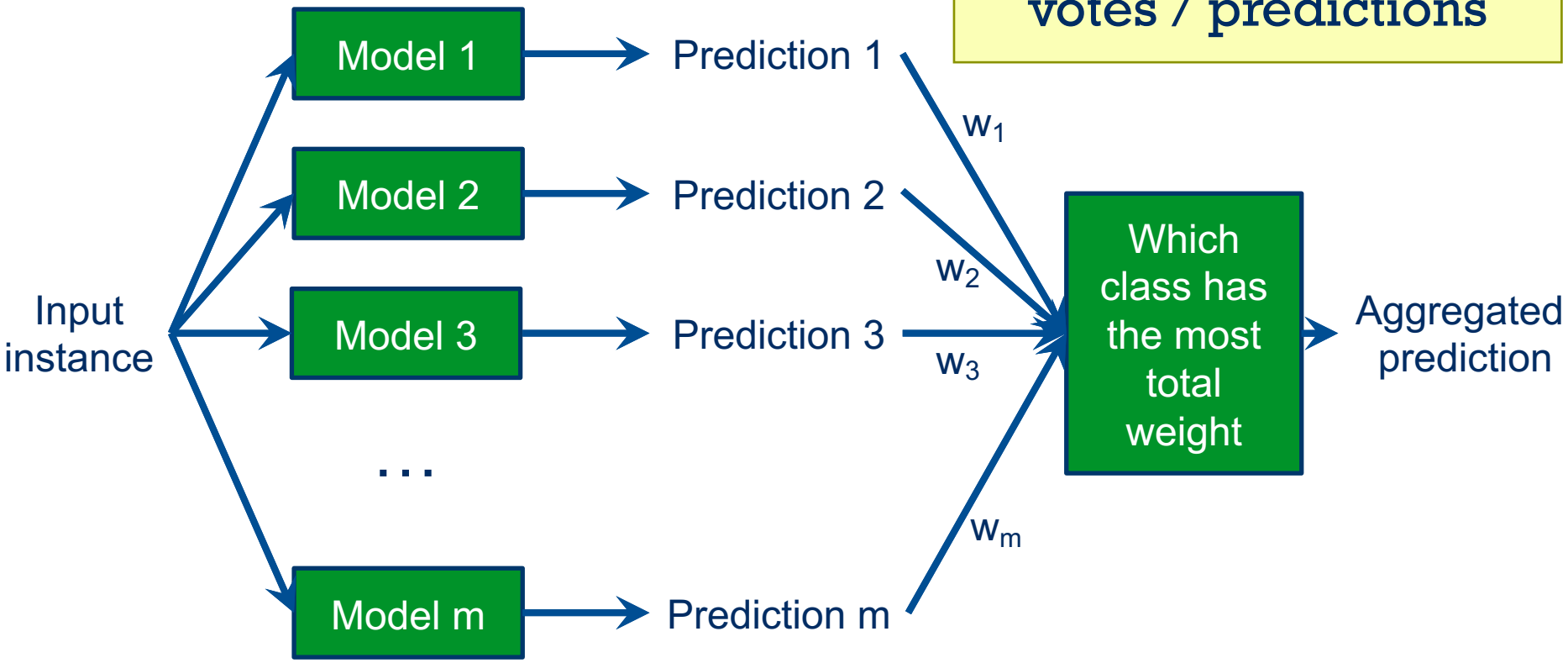
- Homogeneous – all individual models are obtained with the same learning algorithm, on slightly different datasets
- Heterogeneous – individual models are obtained with different algorithms

# Aggregation of predictions

- Classification:
  - Voting
  - Weighted voting

# Weighted voting

$w_1, \ldots, w_m$ - weights for votes / predictions



# Aggregation of predictions

- Classification:
  - Voting
  - Weighted voting
- Regression:
  - Averaging
  - Weighted averaging
- Better models should have higher weights
- How to obtain weights?

# Bayesian model averaging (BMA)

- Suppose we know that one of the M models is the true model but we do not know which
- X – data; T – index of true model; Y – true class; $\hat{y}_1, \ldots, \hat{y}_M$ - predictions of M models

$$P(Y|X) = \sum_{t=1}^{M} P(Y, T=t|X) = \sum_{t=1}^{M} P(Y|T=t, X)P(T=t|X) = \sum_{t=1}^{M} w_t \hat{y}_t$$

- This is weighted averaging of model-specific posterior class probabilities $\hat{y}_t = P(Y|T=t, X)$
- Assuming uniform prior over models, the weights are likelihoods of models:

$$w_t = P(T=t|X) = \frac{P(X|T=t)P(T=t)}{P(X)} \propto P(X|T=t)$$

# Bayesian model averaging (BMA)

- Suppose we know that one of the M models is the true model but we do not know which
- X – data; T – index of true model; Y – true class; $\hat{y}_1, \ldots, \hat{y}_M$ - predictions of M models

An example and further intuition will be given in the lecture on Bayesian machine learning

$P(Y$ $w_t \hat{y}_t$

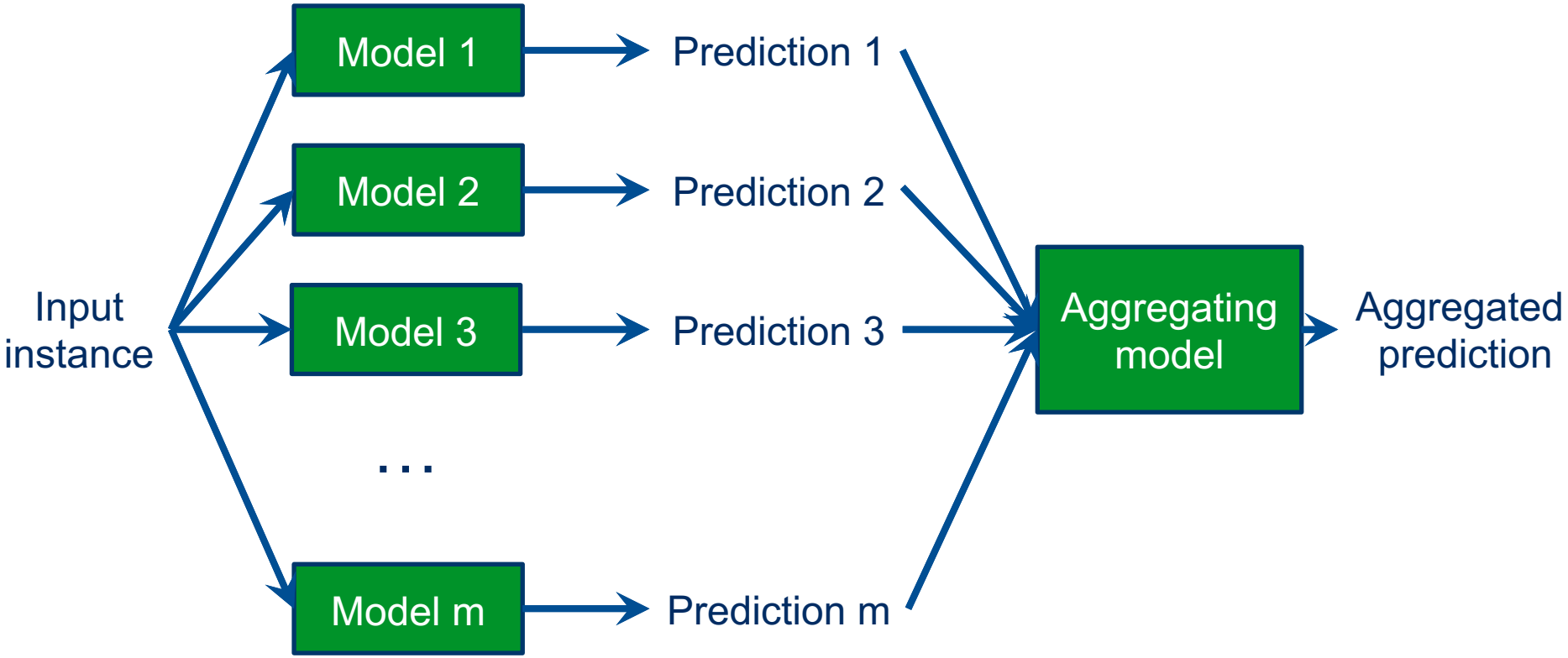
- This is weighted averaging of model-specific posterior class probabilities $\hat{y}_t = P(Y|T=t, X)$
- Assuming uniform prior over models, the weights are likelihoods of models:

$$w_t = P(T=t|X) = \frac{P(X|T=t)P(T=t)}{P(X)} \propto P(X|T=t)$$

# Stacking

- Often we have no way of estimating model likelihoods reliably
- Instead, we can learn the weights in a linear classification task where the individual model outputs are treated as features
- This is known as stacking, because we stack one classifier on top of many individual classifiers

# Stacking


Input instance
Model 1
Model 2
Model 3
...
Model m
Prediction 1
Prediction 2
Prediction 3
Prediction m
Aggregating model
Aggregated prediction

# Lecture 10 – Ensemble methods

- ✓ Why do we need ensemble methods?
- ✓ Bagging
- ✓ Random forest
- ✓ Weighted averaging
- **Boosting**
  - Intuitive explanation
  - AdaBoost algorithm
  - Alternative formulations
  - Interpretations