Machine learning: basi e sue applicazioni

Christian Salvatore
Scuola Universitaria Superiore IUSS Pavia

christian.salvatore@iusspavia.it

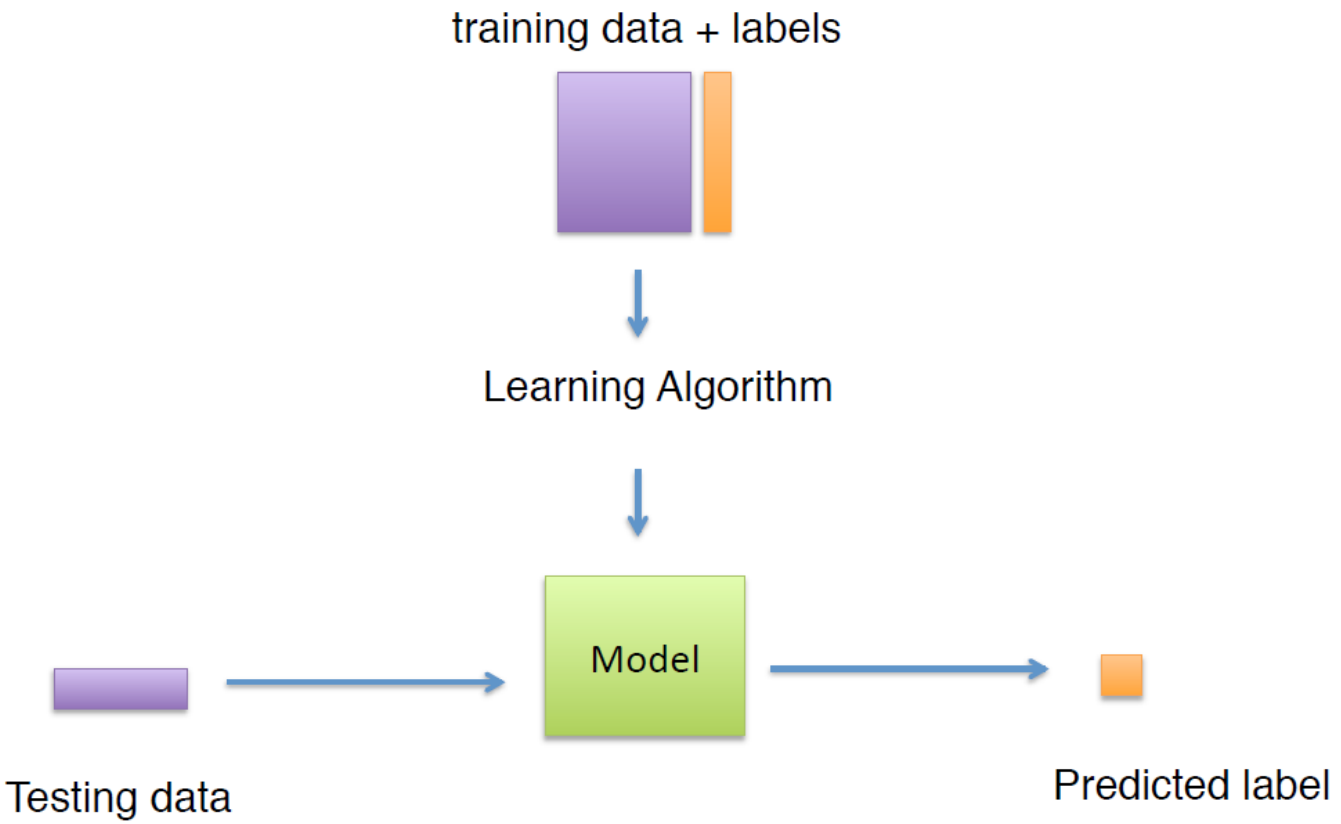
CLASSIFICATION

ENSEMBLES OF CLASSIFIERS

WITH SOME SLIDES FROM PROF. GAVIN BROWN

Ensembles of Classifiers

WITH SOME SLIDES FROM PROF. GAVIN BROWN



Ensembles of Classifiers

WITH SOME SLIDES FROM PROF. GAVIN BROWN

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 12, NO. 10, OCTOBER 1990

Neural Network Ensembles

LARS KAI HANSEN AND PETER SALAMON

***Abstract*—We propose several means for improving the performance and training of neural networks for classification. We use crossvalidation as a tool for optimizing network parameters and architecture. We show further that the remaining residual "generalization" error can be reduced by invoking ensembles of similar networks.**

***Index Terms*—Crossvalidation, fault tolerant computing, neural networks, *N*-version programming.**

I. Introduction

RECENT schemes for training neural networks involving hidden neurons have caused a resurgence of interest in nonalgorithmic supervised learning. A supervised learning scheme is implemented using a *database* which consists of a set of input patterns (a sample from the set of possible inputs) together with the corresponding targets (classifications). The objective of the training is to

performance of the network, we can optimize such performance by varying network characteristics and architecture.

A residual error will typically remain even after optimizing all available network characteristics [6]. To further reduce this error we propose to use a device from fault tolerant computing [7]. We run not a single network but an *ensemble* of networks, each of which have
trained on the same database. The basic idea is to cl
a given input pattern by obtaining a classification
each copy of the network and then using a cons
scheme to decide the collective classification by vo

II. Crossvalidation for Network Optimizati

For supervised learning we employ a database
scribed above including a representative sample of t

International Journal of Forecasting 5 (1989) 559–583
North-Holland

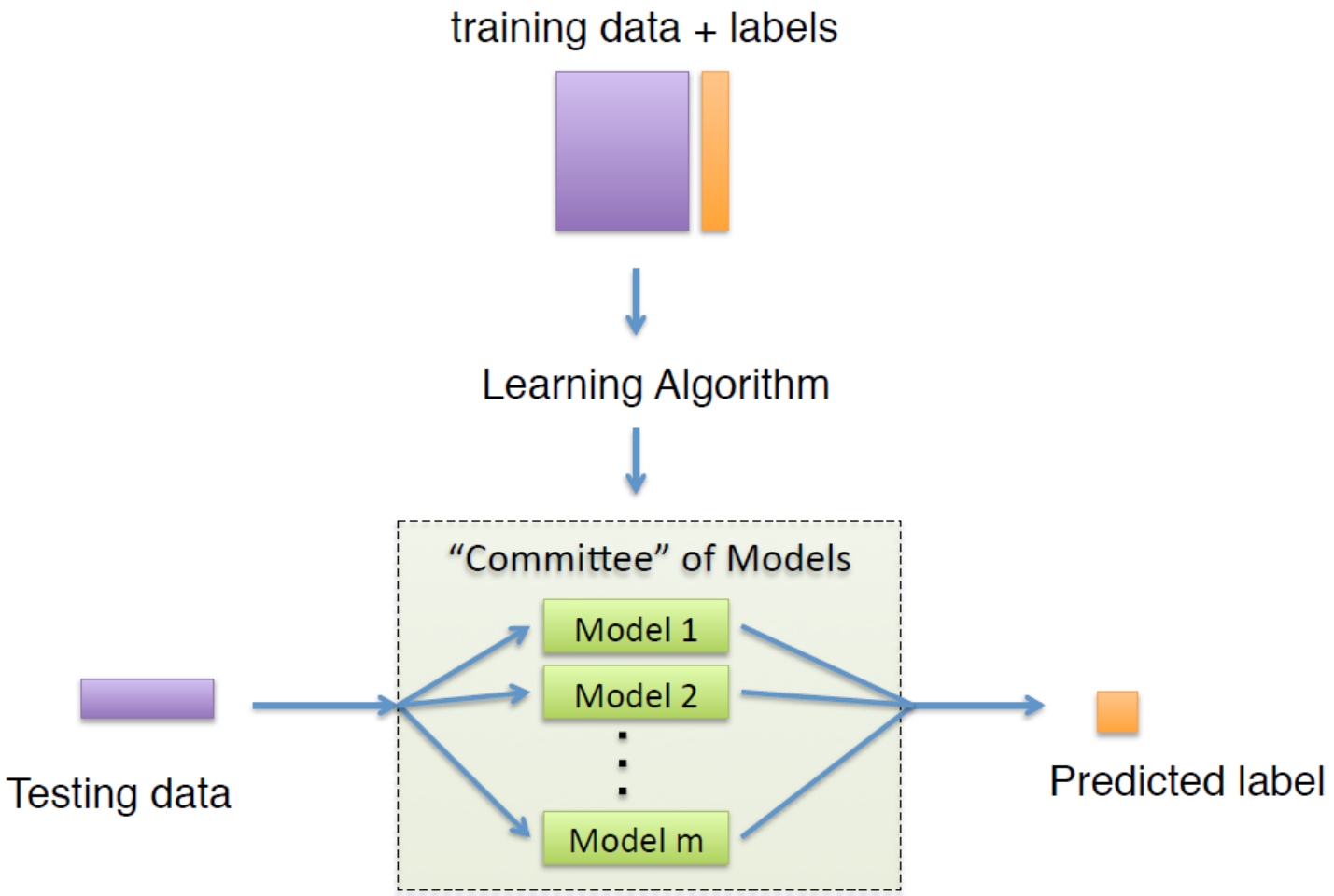
Combining forecasts: A review and annotated bibliography

Robert T. CLEMEN *
College of Business Administration, University of Oregon, Eugene, OR 97403-1208, USA

Abstract: Considerable literature has accumulated over the years regarding the combination of forecasts. The primary conclusion of this line of research is that forecast accuracy can be substantially improved through the combination of multiple individual forecasts. Furthermore, simple combination methods often work reasonably well relative to more complex combinations. This paper provides a review and annotated bibliography of that literature, including contributions from the forecasting, psychology, statistics, and management science literatures. The objectives are to provide a guide to the literature for students and researchers and to help researchers locate contributions in specific areas, both theoretical and applied. Suggestions for future research directions include (1) examination of simple combining approaches to determine reasons for their robustness, (2) development of alternative uses of multiple forecasts in order to

Ensembles of Classifiers

WITH SOME SLIDES FROM PROF. GAVIN BROWN



Ensembles of Classifiers

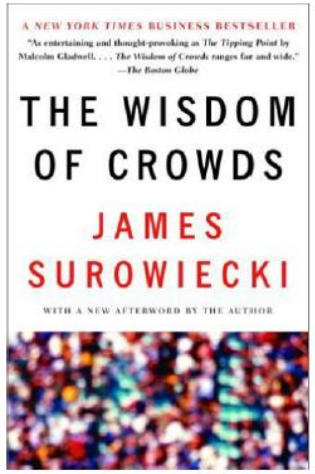
WITH SOME SLIDES FROM PROF. GAVIN BROWN

Combining general predictions?

1906 … county fair in Cornwall, England.

Competition: guess the weight of the cow!

Francis Galton recorded some statistics on the day….



Ensembles of Classifiers

WITH SOME SLIDES FROM PROF. GAVIN BROWN

NATURE [MARCH 7, 1907

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-
ximum
bsolute
bsolute
il 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.

787 guesses.

Truth	1198 lb (~543kg)
Median	1207 lb
Mean	1197 lb

Ensembles of Classifiers

Combining Votes

In 1786 Nicolas de Condorcet (political theorist) asked how do parliaments behave when voting?

…assuming M independent voters…

Marquis de Condorcet

If a single voter has a probability ϵ of making an error, then

$$p(\text{exactly } k \text{ errors}) = \binom{M}{k} \epsilon^k (1-\epsilon)^{(M-k)}$$

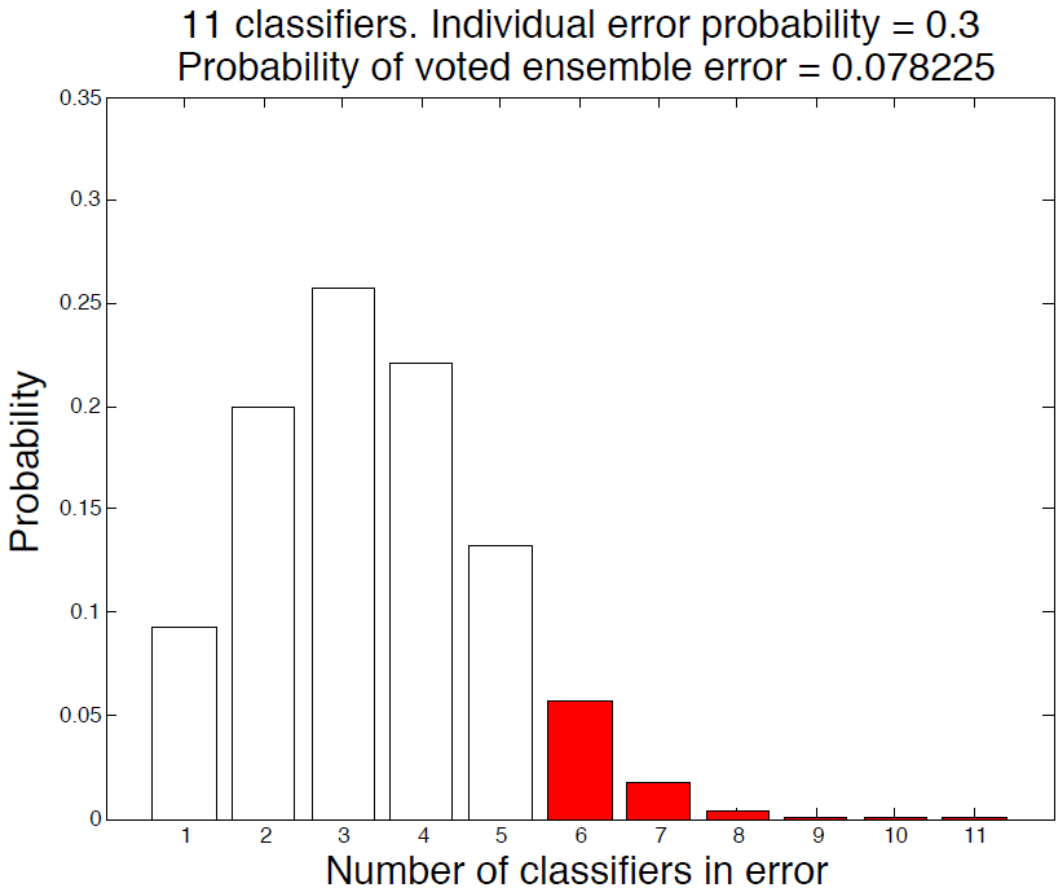
and…

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

Ensembles of Classifiers

WITH SOME SLIDES FROM PROF. GAVIN BROWN

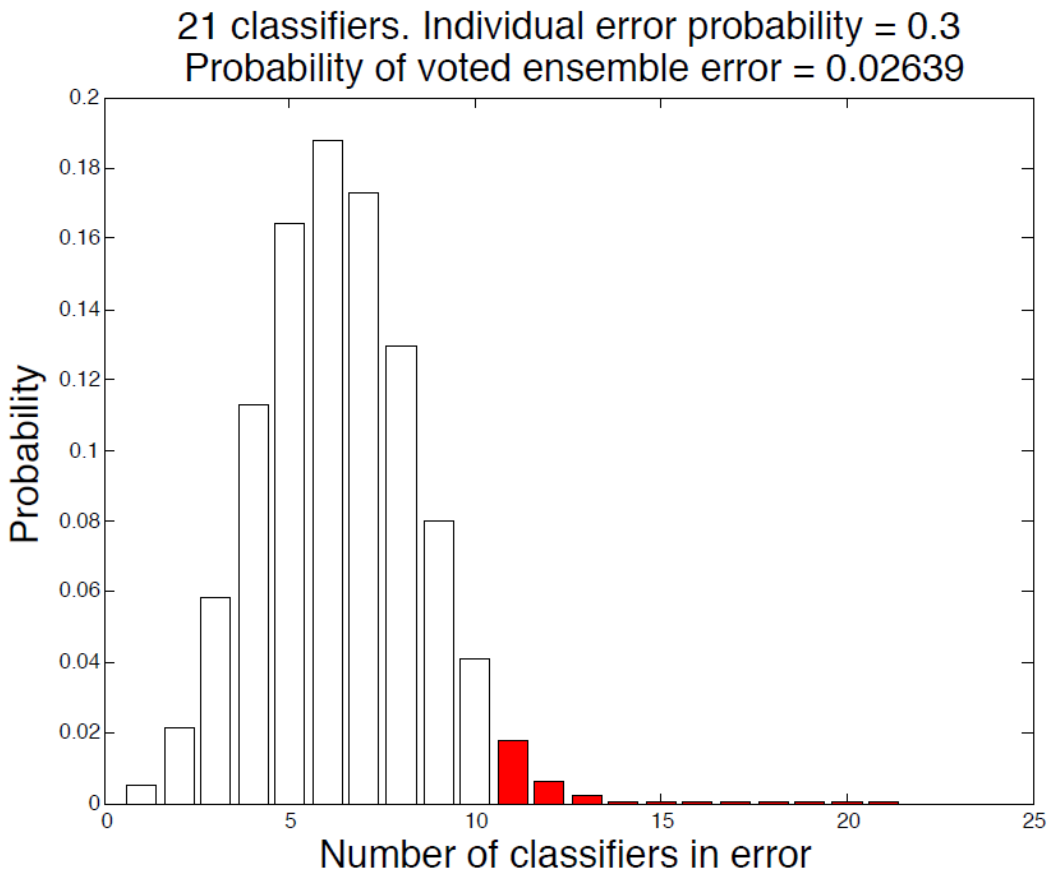
$$p(majority\ vote\ error) = \sum_{k \geq \lceil \frac{M+1}{2} \rceil} \binom{M}{k} \epsilon^k (1-\epsilon)^{(M-k)}$$



Ensembles of Classifiers

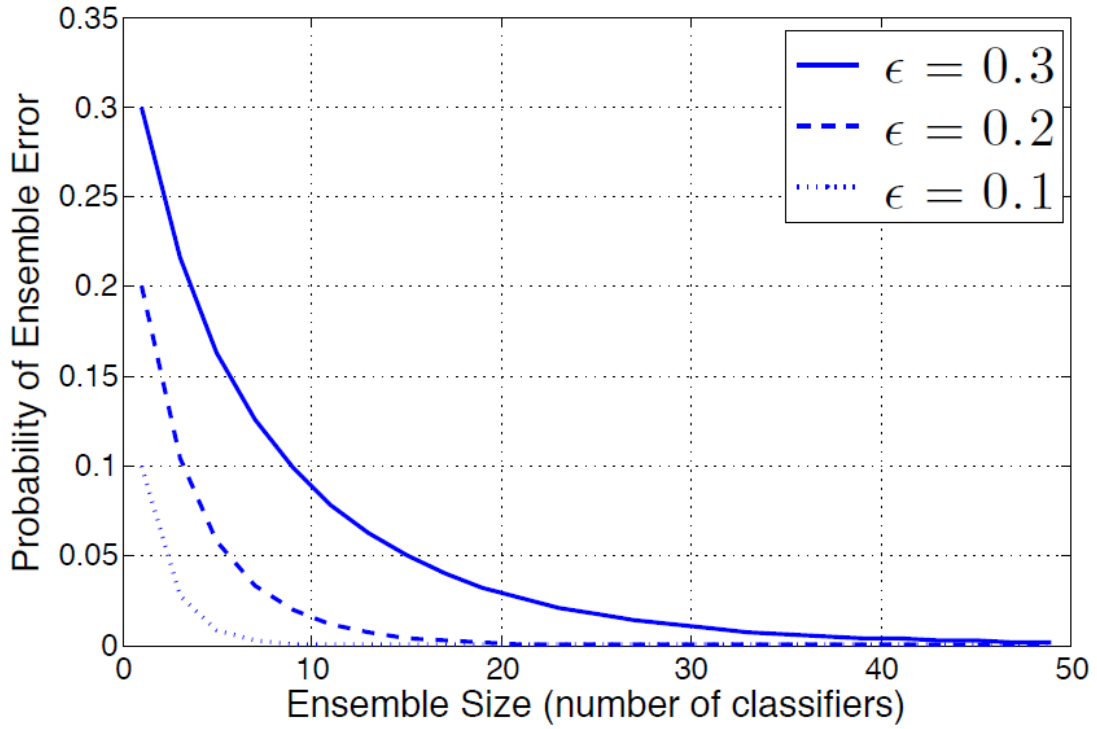
WITH SOME SLIDES FROM PROF. GAVIN BROWN

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



Ensembles of Classifiers

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



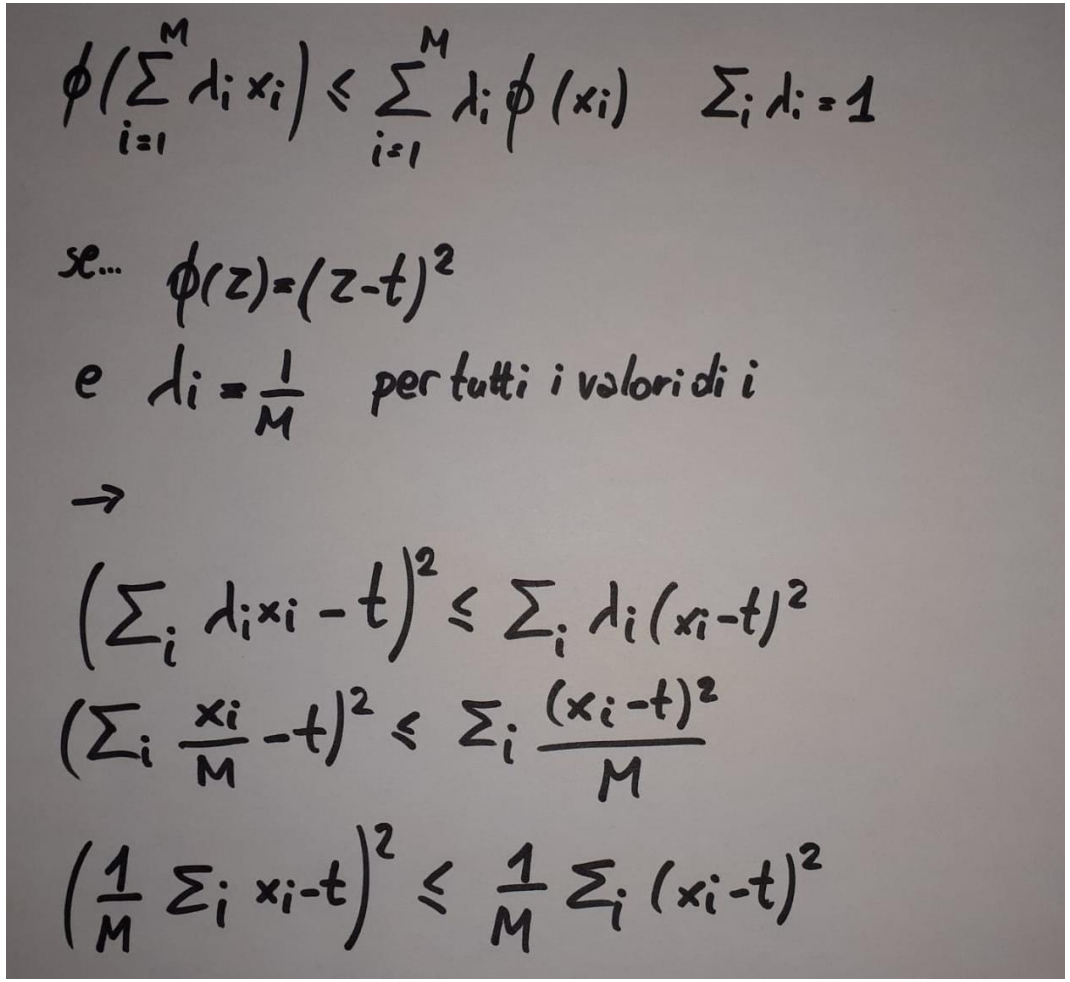
Virtually ZERO error by M = 50 !!

Ensembles of Classifiers

WITH SOME SLIDES FROM PROF. GAVIN BROWN

$$\varphi\left(\mathbb{E}[X]\right) \leq \mathbb{E}[\varphi(X)]$$

Jensen's inequality



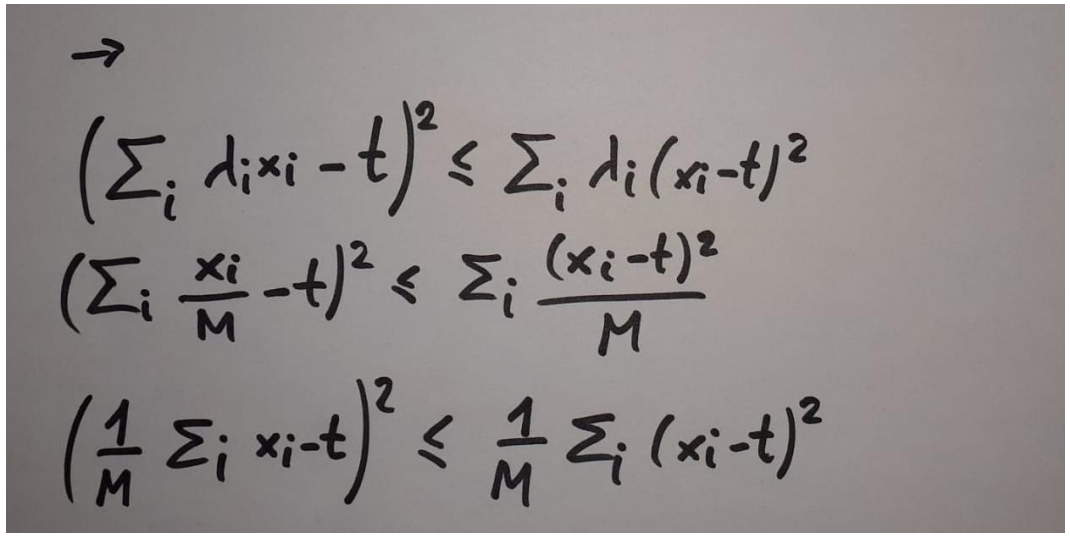
Ensembles of Classifiers

WITH SOME SLIDES FROM PROF. GAVIN BROWN

The squared error of a linear combination of predictions...

....is guaranteed to be less than or equal to...

...the average squared error of the individual predictions.



Ensembles of Classifiers

WITH SOME SLIDES FROM PROF. GAVIN BROWN

Training *different* classifiers . . .

but how “different” the single classifiers should be?

Ensembles of Classifiers

WITH SOME SLIDES FROM PROF. GAVIN BROWN

Training *different* classifiers . . .

but how “different” the single classifiers should be?

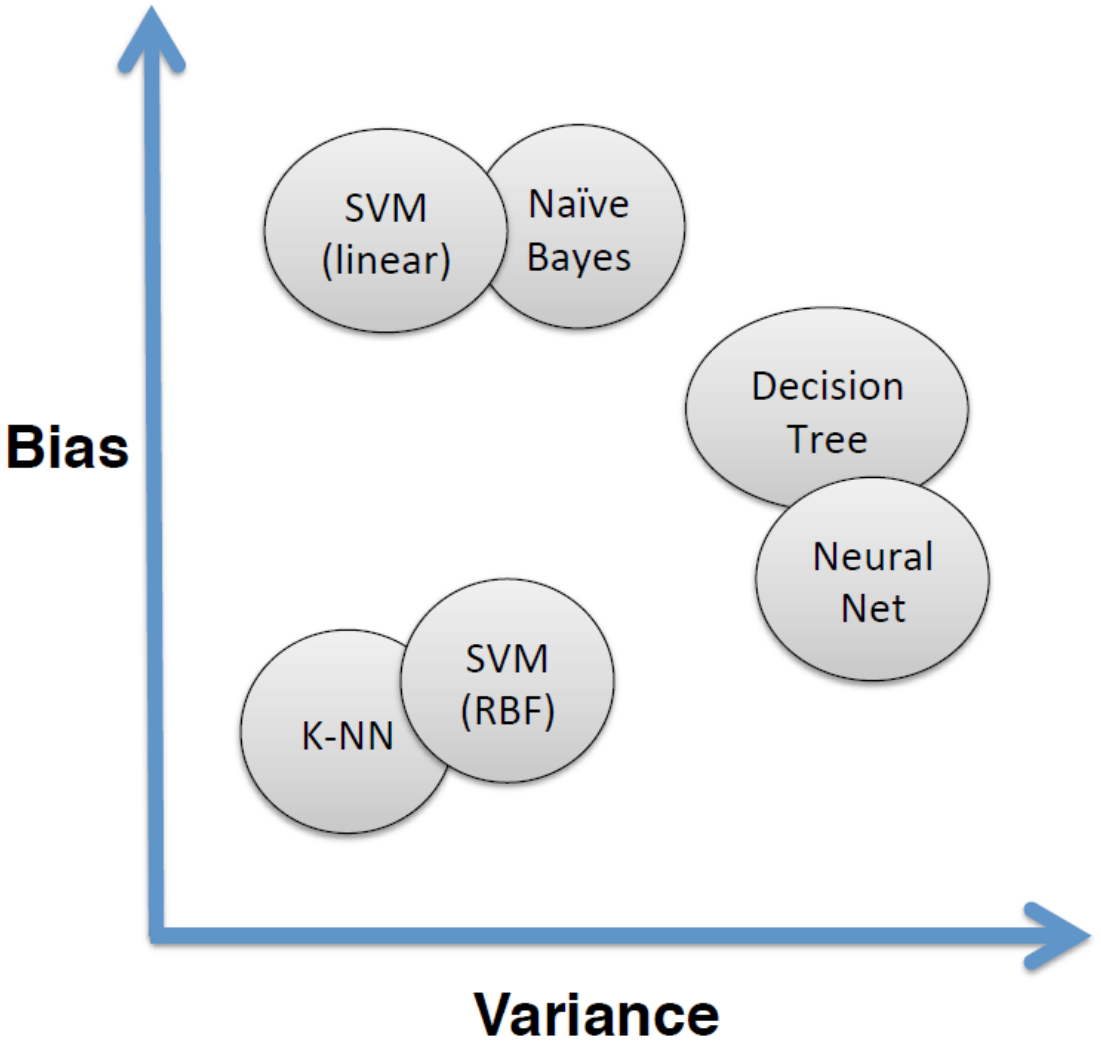
DIFFERENT
(?)

ACCURATE
(?)

INDEPENDENT
(?)

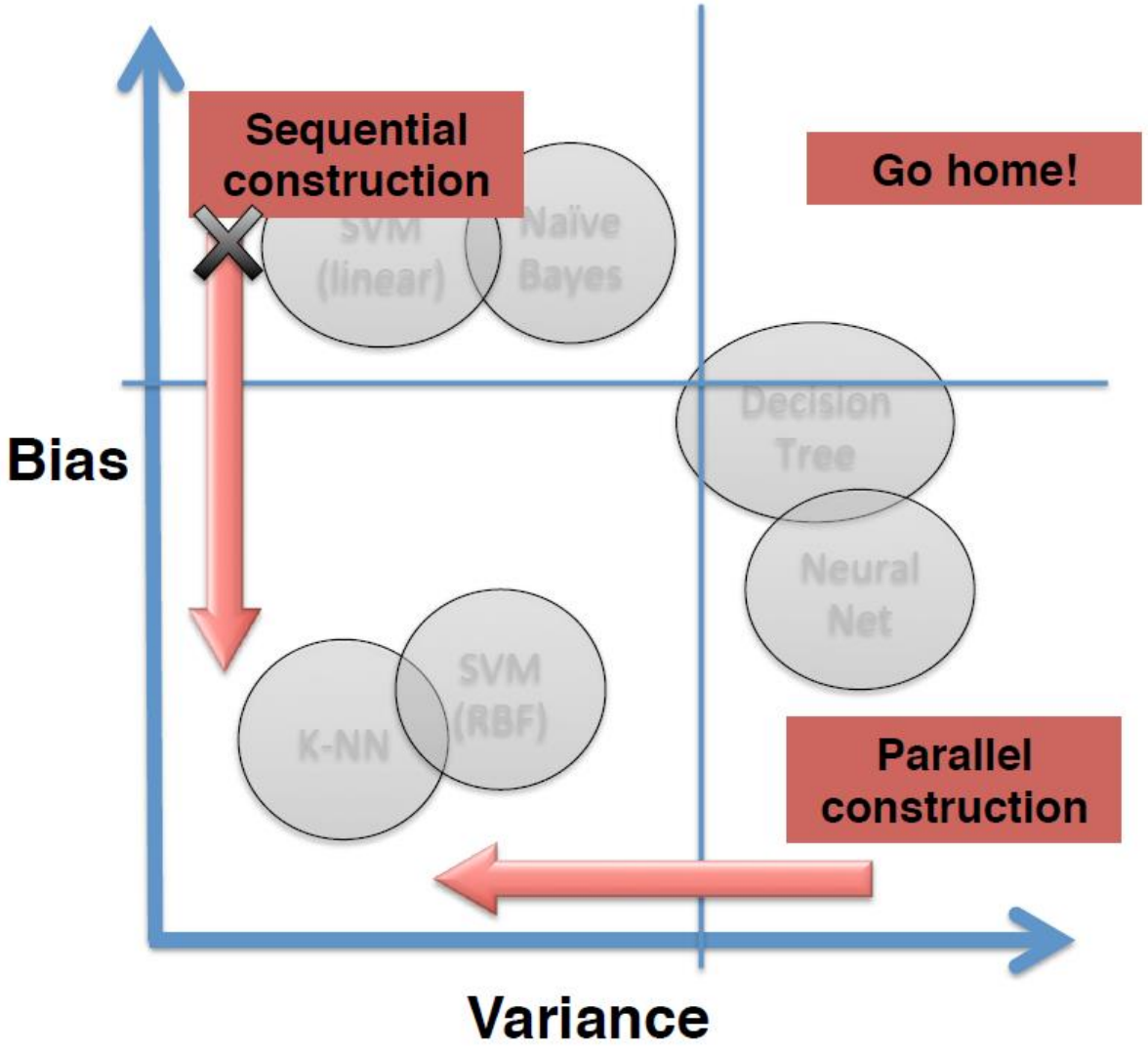
Ensembles of Classifiers

WITH SOME SLIDES FROM PROF. GAVIN BROWN



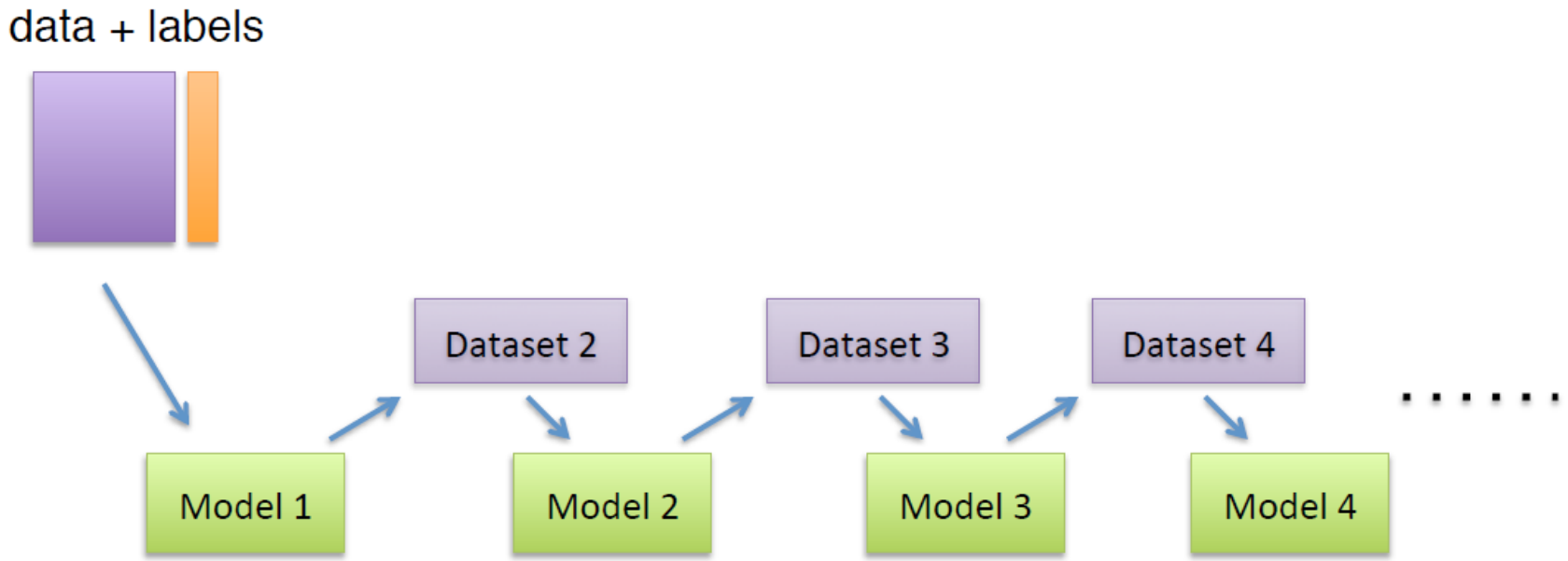
Ensembles of Classifiers

WITH SOME SLIDES FROM PROF. GAVIN BROWN



Ensembles of Classifiers

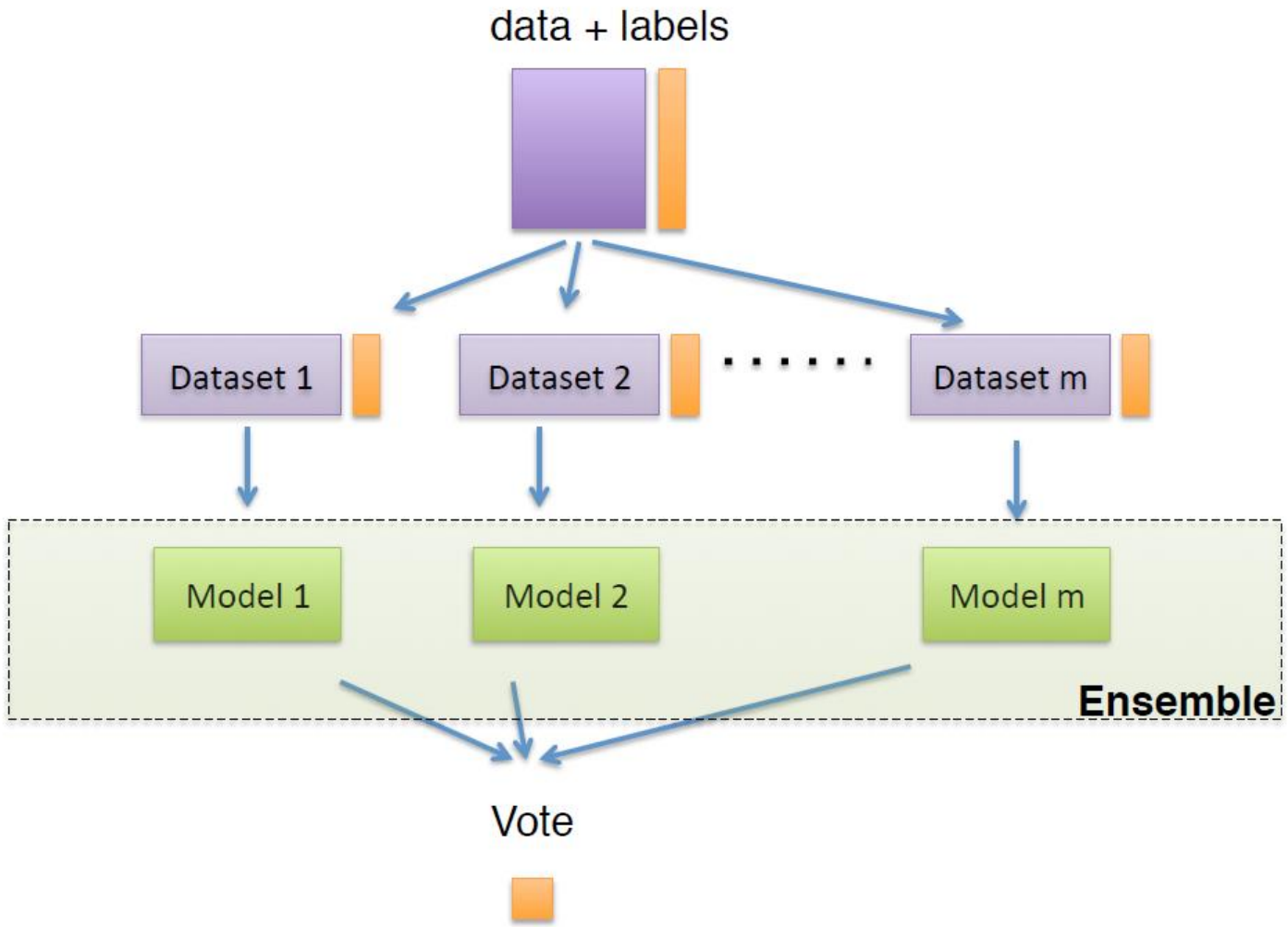
WITH SOME SLIDES FROM PROF. GAVIN BROWN



Each model corrects the mistakes of its predecessor.

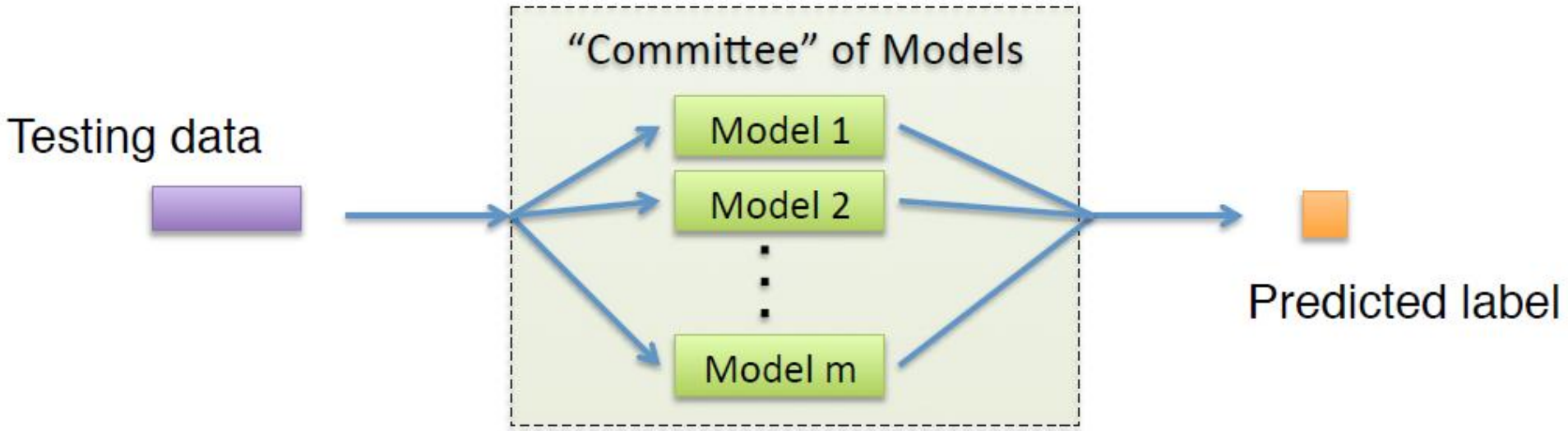
Ensembles of Classifiers

WITH SOME SLIDES FROM PROF. GAVIN BROWN



Ensembles of Classifiers

WITH SOME SLIDES FROM PROF. GAVIN BROWN



Decisions of individuals combined.

At testing phase is the same…

Ensembles of Classifiers

WITH SOME SLIDES FROM PROF. GAVIN BROWN

Which are the most popular methods to build "ensemble" classifiers?

BAGGING

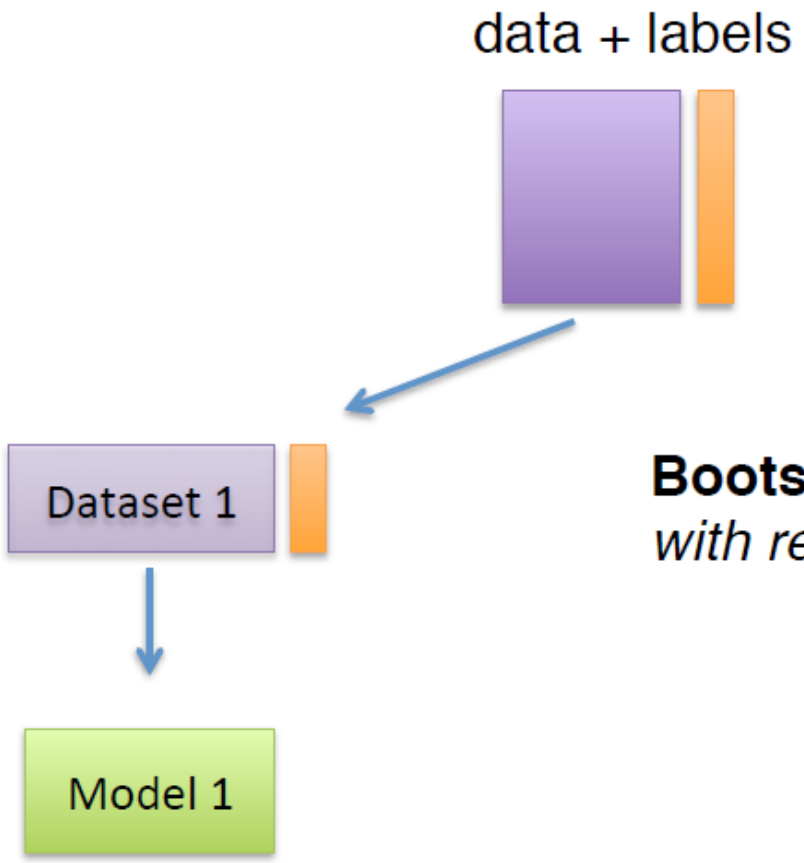
BOOSTING

STACKING

BLENDING

Ensembles of Classifiers | Bagging

BAGGING = BOOTSTRAP AGGREGATING



Bootstrap: Sample N examples,
with replacement, from the original N.

Ensembles of Classifiers | Bagging

Ensembles of Classifiers | Bagging

Probability of including any given example in a bootstrap:

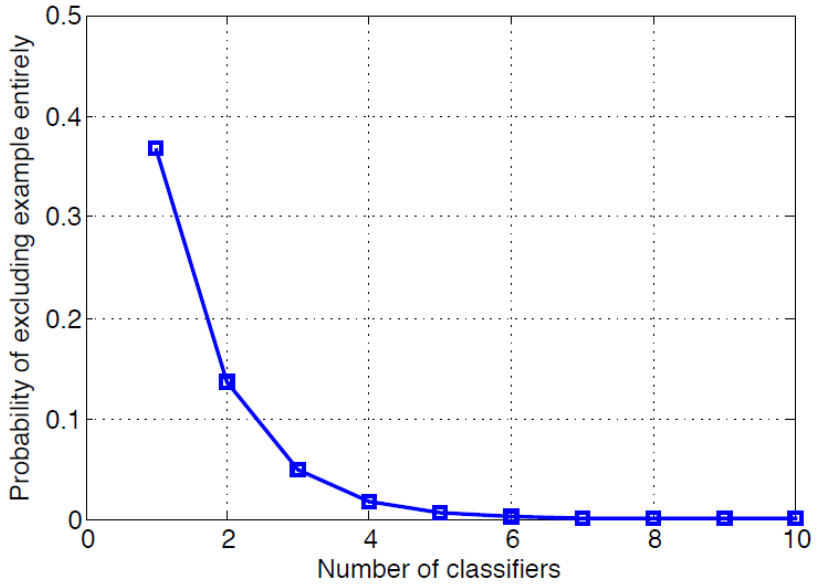
$$p = 1 - (1 - \frac{1}{N})^N$$

$$\lim_{N \to \infty} \left\{ 1 - (1 - \frac{1}{N})^N \right\} = 1 - e^{-1} \approx 0.6321$$

~63% of the data included.

… 37% left out!?

PROBABILITY OF EXCLUDING A SAMPLE FROM THE WHOLE ENSEMBLE



Ensembles of Classifiers | Bagging

Bagging (input training data+labels T, number of models M)

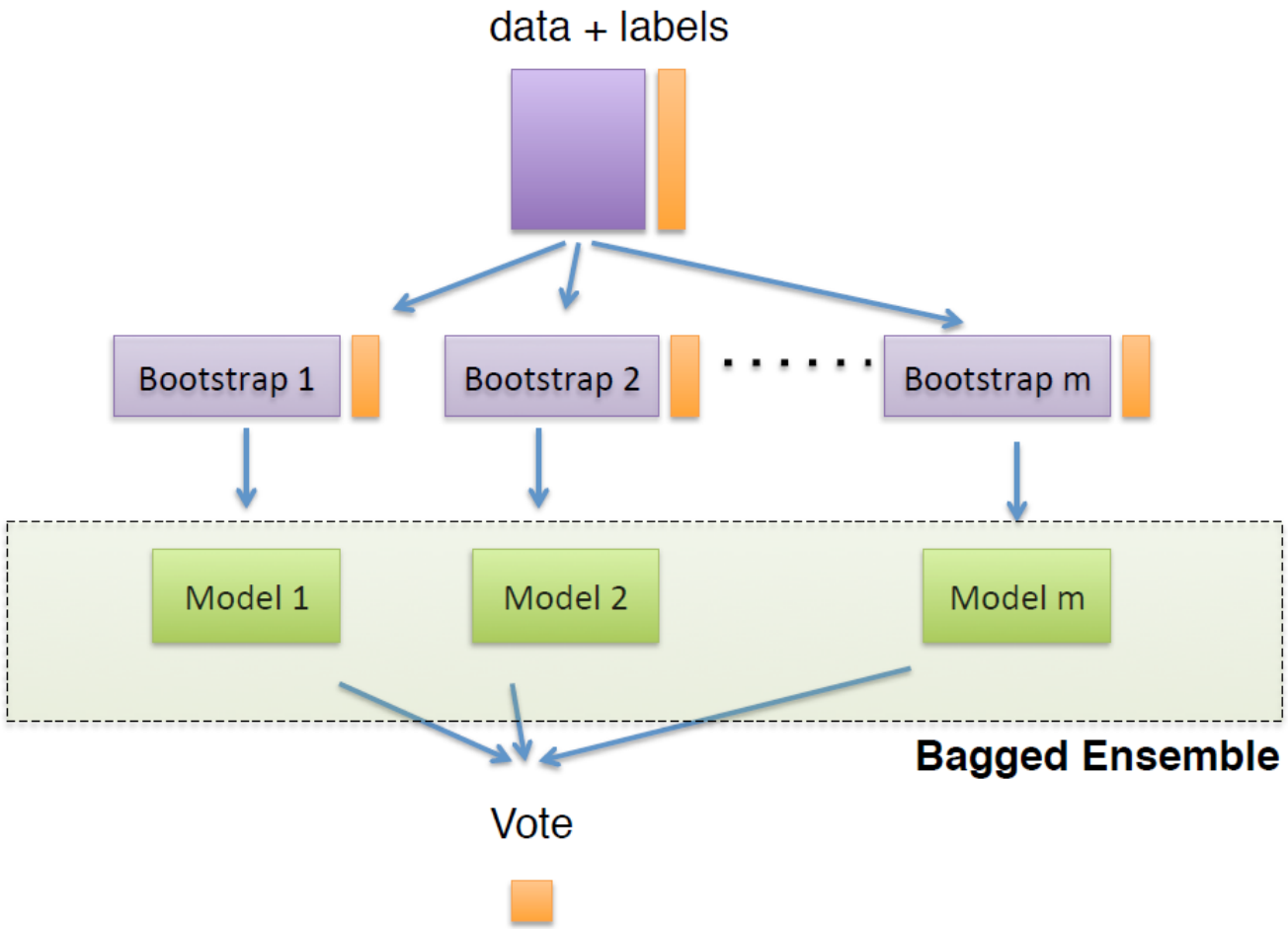
---

for $j = 1$ to M **do**
    Take a *bootstrap* sample T' from T
    Build a model using T'.
    Add the model to the set.
end for
return set of models

For a test point $\mathbf{x}$, get a response from each model, and take a majority vote.

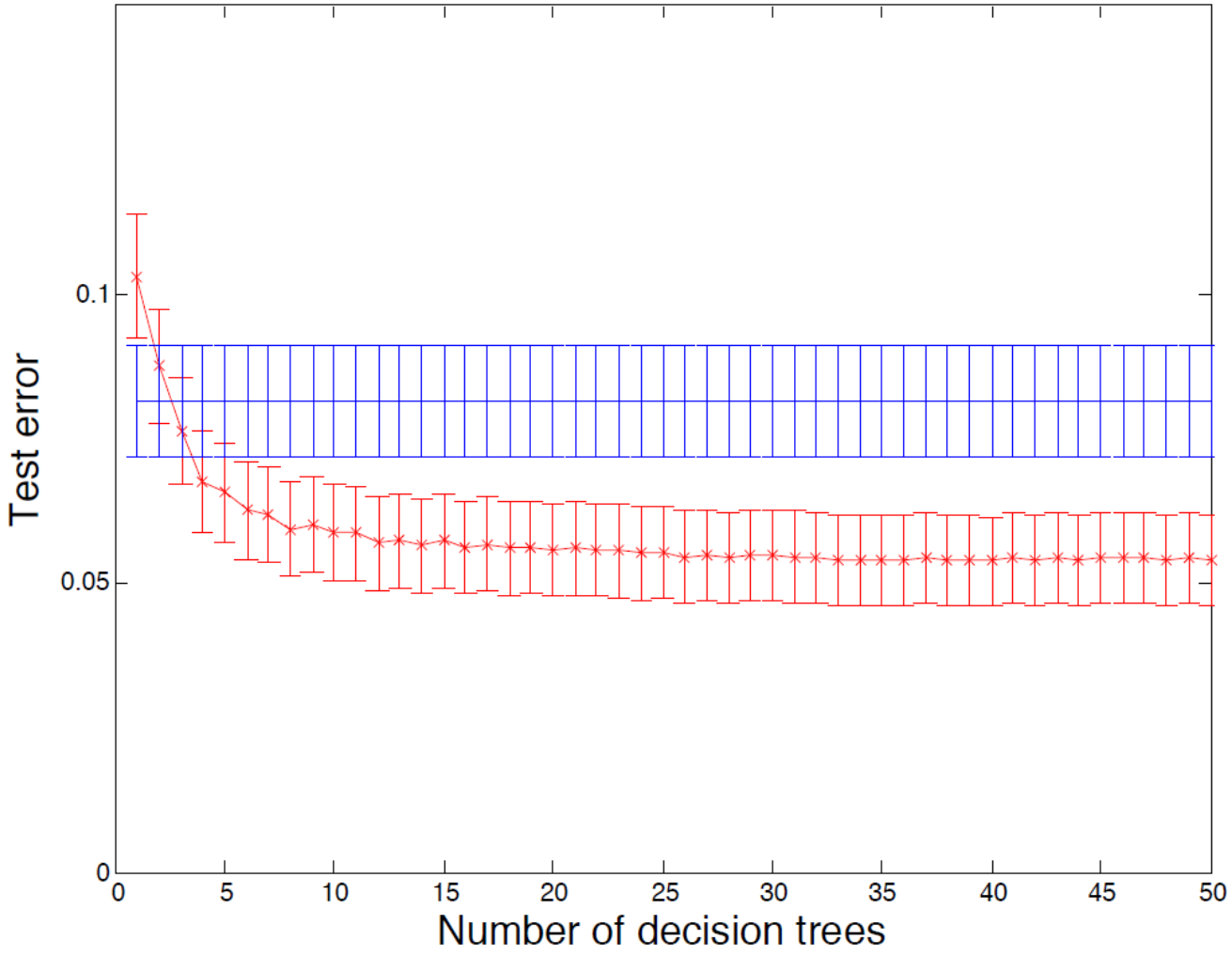
Breiman, 1996
(who will implement random forests 4 years later…)

Ensembles of Classifiers | Bagging



Ensembles of Classifiers | Bagging

WITH SOME SLIDES FROM PROF. GAVIN BROWN



Ensembles of Classifiers | Bagging

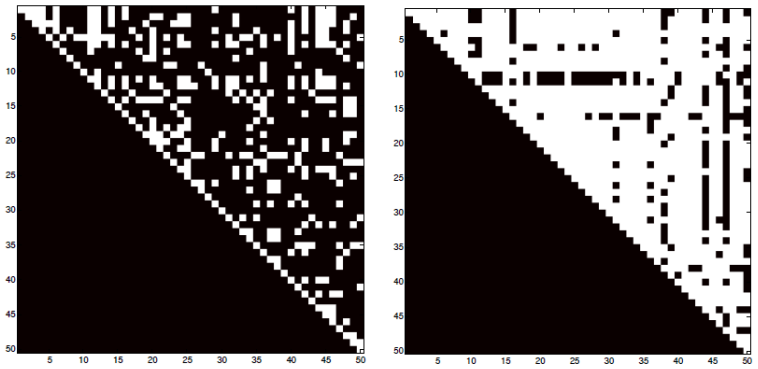
Training *different* classifiers . . .

but how "different" the single classifiers should be?

DIFFERENT
(!)

ACCURATE
(~)

INDEPENDENT
(?)



Dependencies between 50 bagged classifiers – CART (left), Gaussian Naive Bayes (right).

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

Ensembles of Classifiers | Bagging

A note on model stability

Some models are almost completely unaffected by bootstrapping (similar to each other on test data)

STABLE MODELS are not the first choice for parallel ensemble methods

Neural Networks
Logistic Regression
kNN
SVM
Decision Trees

STABLE

UNSTABLE

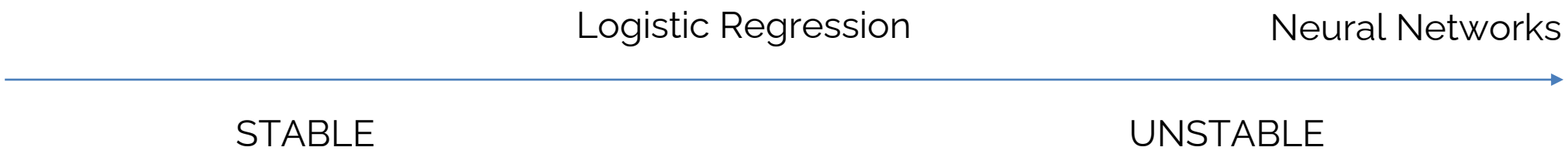
Ensembles of Classifiers | Bagging

A note on model stability

Some models are almost completely unaffected by bootstrapping (similar to each other on test data)

STABLE MODELS are not the first choice for parallel ensemble methods

kNN
SVM
Decision Trees

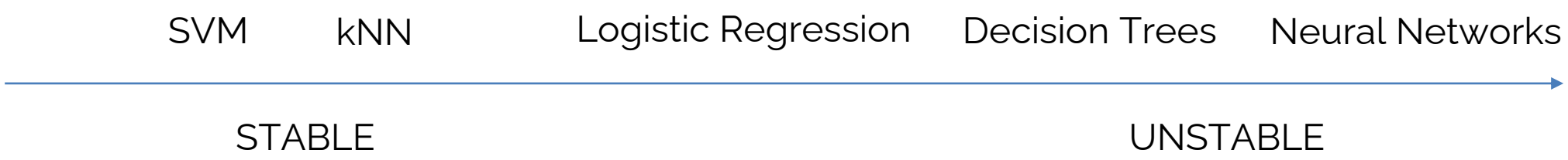


Ensembles of Classifiers | Bagging

A note on model stability

Some models are almost completely unaffected by bootstrapping (similar to each other on test data)

STABLE MODELS are not the first choice for parallel ensemble methods



Ensembles of Classifiers | Stacking

Stacking or Stacked Generalization

"Given multiple machine learning models that are skillful on a problem, but in different ways, how do you choose which model to use (trust)?"

Use a second-level (meta) machine-learning model that learns when to use (trust) each model in the ensemble

Ensembles of Classifiers | Stacking

Architecture of a stacking model:

- two or more base models (often referred to as LEVEL-0 models), which learn to discriminate training data at a single-classifier level and whose predictions are computed and recorded (to be used at level 1)

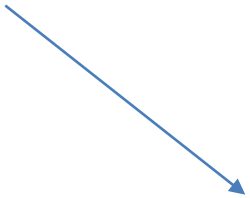
- one meta-model (LEVEL 1) that combines the predictions of the base models (referred to as a LEVEL-1 model) by learning how to best combine LELEVL-0 models (or their predictions)

Ensembles of Classifiers | Stacking

Architecture of a stacking model:

- two or more base models (often referred to as LEVEL-0 models), which learn to discriminate training data at a single-classifier level and whose predictions are computed and recorded (to be used at level 1)

- one meta-model (LEVEL 1) that combines the predictions of the base models (referred to as a LEVEL-1 model) by learning how to best combine LELEVL-0 models (or their predictions)



SHOULD BE:
1. sufficiently accurate
2. Sufficiently different (assumptions about the predictive task)
3. sufficiently independent from each other (uncorrelated in their predictions)

Ensembles of Classifiers | Stacking

Architecture of a stacking model:

- two or more base models (often referred to as LEVEL-0 models), which learn to discriminate training data at a single-classifier level and whose predictions are computed and recorded (to be used at level 1)

- one meta-model (LEVEL 1) that combines the predictions of the base models (referred to as a LEVEL-1 model) by learning how to best combine LELEVL-0 models (or their predictions)

LEARN classifiers' combination through cross validation (or similar)

SHOULD BE:
1. sufficiently accurate
2. Sufficiently different (assumptions about the predictive task)
3. sufficiently independent from each other (uncorrelated in their predictions)

Ensembles of Classifiers | Blending

WITH SOME SLIDES FROM PROF. GAVIN BROWN

> *Blending is a word introduced by the Netflix winners. It is very close to stacked generalization, but a bit simpler and less risk of an information leak. […] With blending, instead of creating out-of-fold predictions for the train set, you create a small holdout set of say 10% of the train set. The stacker model then trains on this holdout set only.*

— Kaggle Ensemble Guide, MLWave, 2015.

The BellKor 2008 Solution to the Netflix Prize

Robert M. Bell
AT&T Labs - Research
Florham Park, NJ

Yehuda Koren
Yahoo! Research
Haifa, Israel

Chris Volinsky
AT&T Labs - Research
Florham Park, NJ

BellKor@research.att.com

1. Introduction

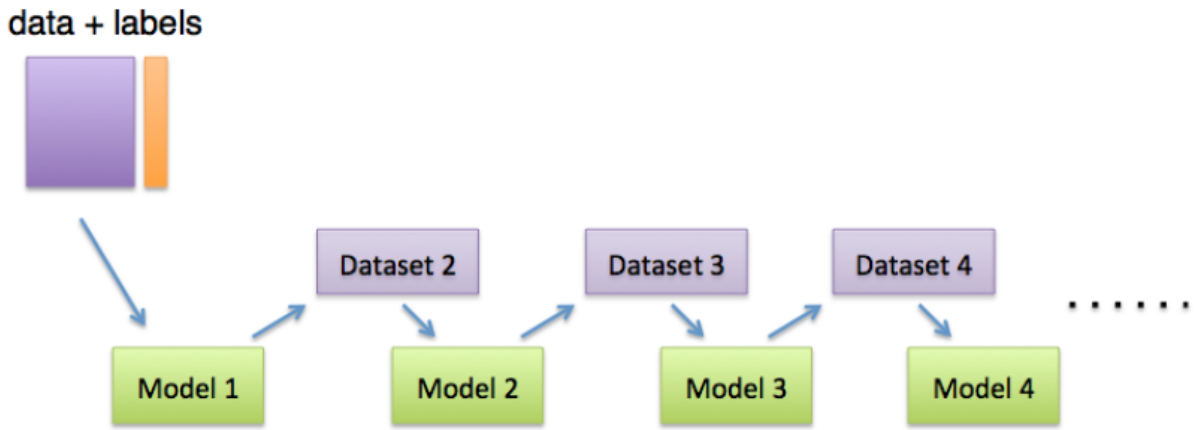
Our RMSE=0.8643[2] solution is a linear blend of over 100 results. Some of them are new to this year, whereas many others belong to the set that was reported a year ago in our 2007 Progress Prize report [3]. This report is structured accordingly. In Section 2 we detail methods new to this year. In general, our view is that those newer methods deliver a superior performance compared to the methods we used a year ago. Throughout the description of the methods, we highlight the specific predictors that participated in the final blended solution. Nonetheless, the older methods still play a role in the blend, and

- **Blending**: Stacking-type ensemble where the meta-model is trained on predictions made on a holdout dataset.
- **Stacking**: Stacking-type ensemble where the meta-model is trained on out-of-fold predictions made during k-fold cross-validation.

Ensembles of Classifiers | Boosting

WITH SOME SLIDES FROM PROF. GAVIN BROWN

Define a distribution over the training set, $D_1(i) = \frac{1}{N}, \forall i$.
for $t = 1$ to T **do**
    Build a model h_t from the training set, using distribution D_t.
    Update D_{t+1} from D_t:
        Increase the weight on examples that h_t incorrectly classifies.
        Decrease the weight on examples that h_t correctly classifies.
end for
For a new testing point $(\mathbf{x}', y')$, we take a weighted majority vote from $\{h_1, ...h_T\}$.



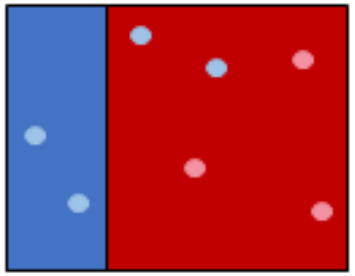
Ensembles of Classifiers | Boosting

WITH SOME SLIDES FROM PROF. GAVIN BROWN

ADABOOST

AdaBoost:

- Combining **weak learners** (decision trees)
- Assigning **weights to incorrect values**
- **Sequential tree growing** considering past mistakes



Results of tree 1

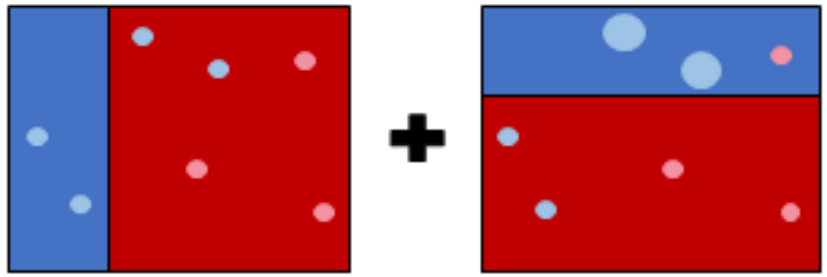
Ensembles of Classifiers | Boosting

WITH SOME SLIDES FROM PROF. GAVIN BROWN

ADABOOST

AdaBoost:

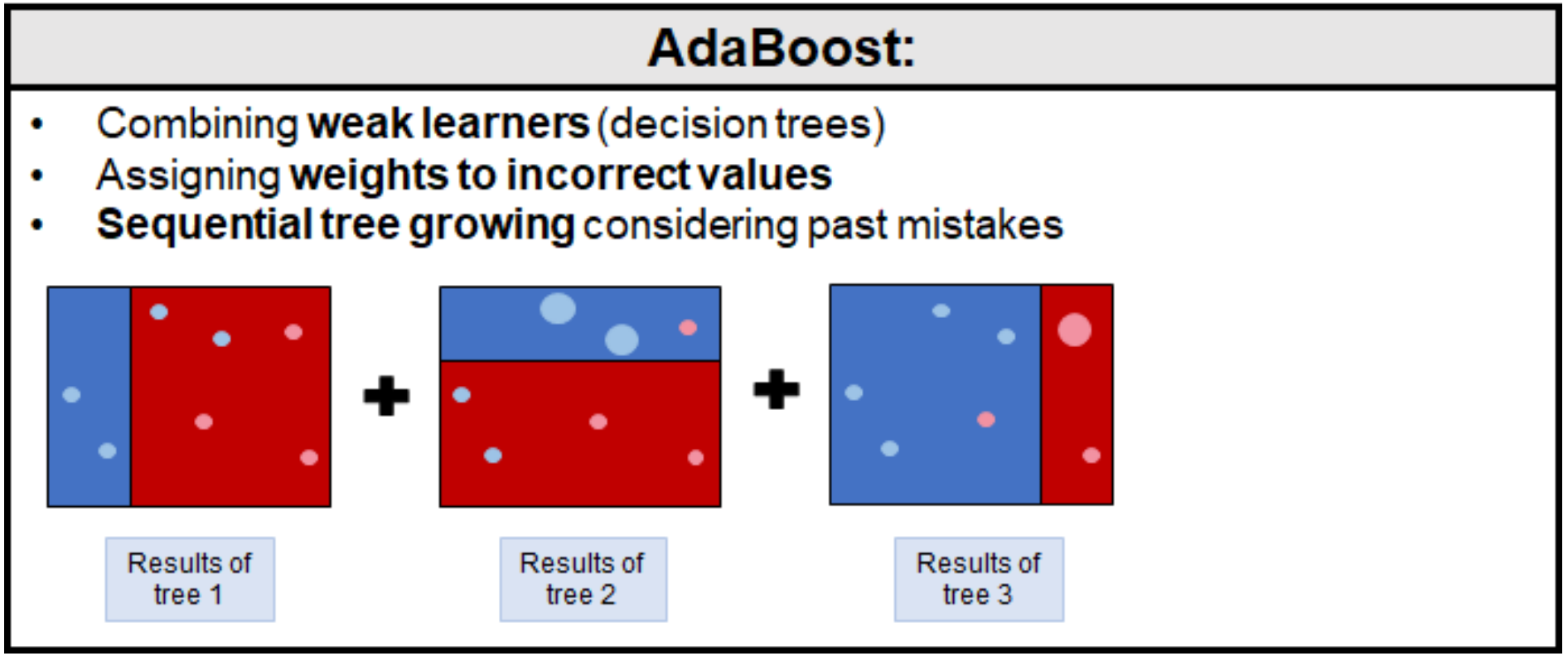
- Combining **weak learners** (decision trees)
- Assigning **weights to incorrect values**
- **Sequential tree growing** considering past mistakes



Results of tree 1

Results of tree 2

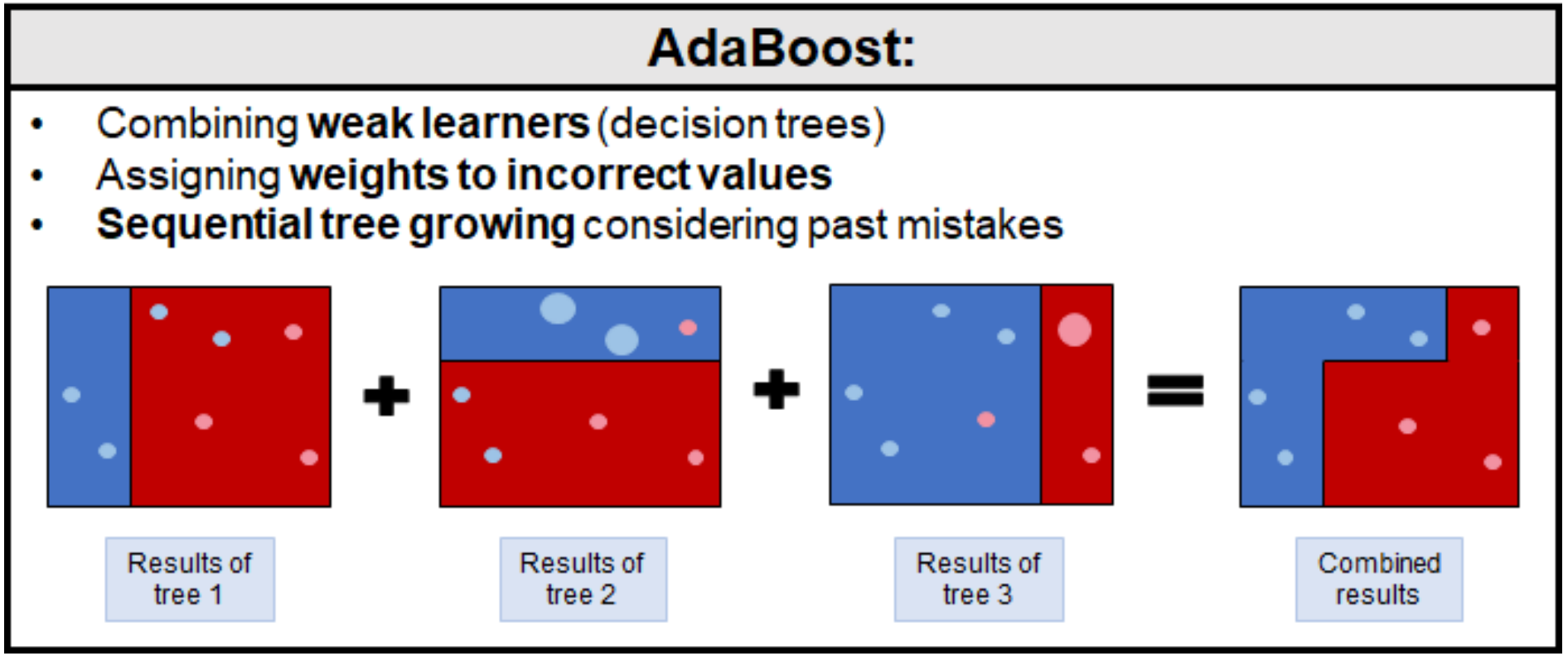
ADABOOST


AdaBoost:
• Combining weak learners (decision trees)
• Assigning weights to incorrect values
• Sequential tree growing considering past mistakes
Results of tree 1
Results of tree 2
Results of tree 3

Ensembles of Classifiers | Boosting

WITH SOME SLIDES FROM PROF. GAVIN BROWN

ADABOOST



ADABOOST

Define a distribution over the training set, $D_1(i) = \frac{1}{N}, \forall i.$
for $t = 1$ to T **do**
Build a classifier h_t from the training set, using distribution D_t.
Set $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$
Update D_{t+1} from D_t :
Set $D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$
end for

$$H(x') = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x') \right)$$

$$where \quad Z_t = \sum_i D_t(i) e^{-y_i \alpha_t h_t(x_i)}$$

Ensembles of Classifiers | Boosting

ADABOOST

Define a ain
for $t = 1$
Build a classifier h_t from the tra
Set $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$
Update D_{t+1} from D_t :
Set $D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i}}{Z_t}$
end for

Error rate

Weight Update

$$H(x') = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x') \right)$$

$$E_1 = \frac{1}{N} \sum_{i=1}^{N} e^{-y_i h_1(x_i)}$$

Exponential loss

Sum of the exponential loss on each data point

$$E_2 = \sum_{i=1}^{N} \underbrace{\frac{1}{N} e^{-y_i \alpha_1 h_1(x_i)}}_{constant} e^{-y_i \alpha_2 h_2(x_i)}$$

$$E_2 = \sum_{i=1}^{N} w_2(i) e^{-y_i \alpha_2 h_2(x_i)}$$

$$where \quad Z_t = \sum_i D_t(i) e^{-y_i \alpha_t h_t(x_i)}$$

Ensembles of Classifiers | Boosting

ADABOOST

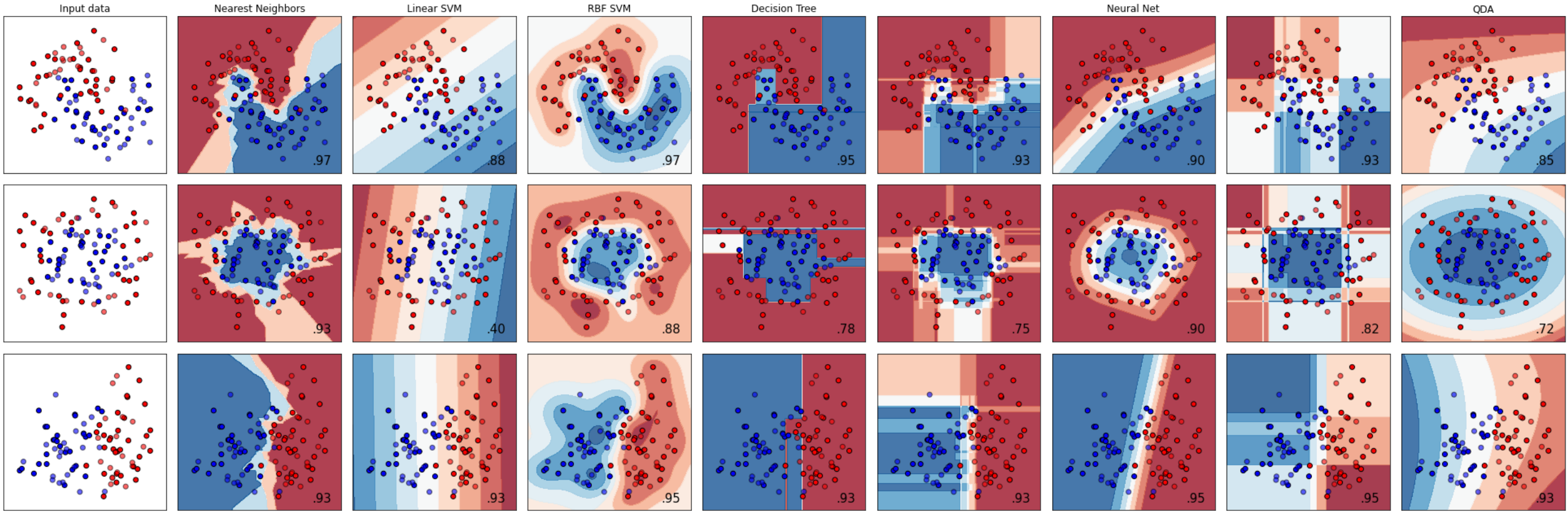
Good performance with non-noisy datasets

Controversial results with noisy data

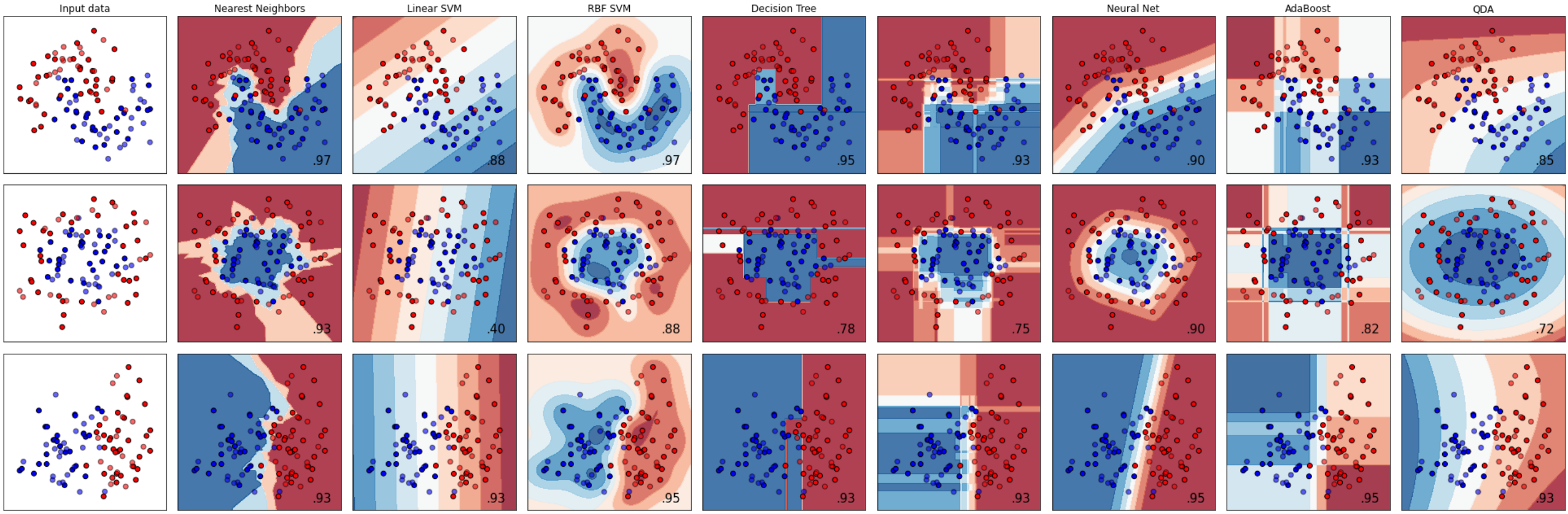
Low number of hyperparameters

Not optimized for computational costs in terms of time

Which is which | Decision Function



Which is which | Decision Function



Ensembles of Classifiers | To Ensemble or Not?

WITH SOME SLIDES FROM PROF. GAVIN BROWN

To Ensemble or Not Ensemble: When does End-To-End Training Fail?

Andrew Webb[1], Charles Reynolds[1], Wenlin Chen[1], Henry Reeve[2], Dan Iliescu[3], Mikel Luján[1], and Gavin Brown[1]

[1] University of Manchester, UK
[2] University of Bristol, UK
[3] University of Cambridge, UK

Abstract. End-to-End training (E2E) is becoming more and more popular to train complex Deep Network architectures. An interesting question is whether this trend will continue—are there any clear failure cases for E2E training? We study this question in depth, for the specific case of E2E training an *ensemble* of networks. Our strategy is to blend the gradient smoothly in between two extremes: from independent training of the networks, up to to full E2E training. We find clear failure cases, where overparameterized models *cannot be trained E2E*. A surprising result is that the optimum can sometimes lie in between the two, neither an ensemble or an E2E system. The work also uncovers links to Dropout, and raises questions around the nature of ensemble diversity and multi-branch networks.

CLASSIFICATION

RANDOM FOREST

Random Forest

Ideato da Leo Breiman nel 2001, è un classificatore multiplo in cui i singoli classificatori sono classification tree (alberi di classificazione) (estimatori)

- Genera alberi multipli ma a partire sempre dallo stesso training set

- Per la crescita dell'albero si sceglie ad ogni livello la feature che meglio separa le classi (foglie) e si determina la soglia di suddivisione

- Per la classificazione di un nuovo pattern (test set) si visita l'albero, e una volta giunti a una foglia, si classifica il pattern sulla base della classe più comune nel nodo tra i pattern del training set

- La decisione avviene mediante fusione dei classificatori (e.g. majority vote rule)

Random Forest | Bagging

- Per ogni nodo la scelta della feature migliore su cui partizionare non è fatta sull'intero insieme delle d feature (dimensionalità dei pattern), ma su un sottoinsieme random di d' feature:

Valore tipico d' = sqrt (d)

-In assenza di questo accorgimento (noto anche come feature bagging) molti alberi sceglierebbero con elevata probabilità le stesse variabili (quelle più discriminanti).

-Allo stesso tempo anche il training non è fatto sull'intero insieme degli n pattern, ma su un sottoinsieme random di n' pattern.

Valore tipico n' = 2/3 n

Bagging = Bootstrap AGGregatING

Random Forest | Bagging

- Random Forest opera simultaneamente due tipi di bagging: uno sui pattern del training set e uno sulle features.

- Le prestazioni possono essere stimate con validation set separato (k-fold cross validation) o, grazie al bagging, con tecnica Out-Of-Bag (OOB) che non richiede validation set separato. Infatti ciascun pattern x può essere utilizzato per stimare le prestazioni a partire dai soli tree nel cui training x non è stato coinvolto.

Random Forest | Importance

Mean decrease in the Gini index.
Misura analoga a quella usata negli alberi di decisione: per ogni albero e per ogni variabile i, si valuta ogni nodo in cui lo split coinvolge tale variabile, calcolando la differenza dell'indice di Gini (impurità) prima e dopo la suddivisione (nodo madre e nodi figli), pesandola per il numero di campioni nel nodo. Quindi si fa la media sugli alberi.

Più alti i valori, maggiore è la rilevanza della feature.

Random Forest

Random Forests (input training data+labels T, number of trees M)

for $j = 1$ to M **do**
Take a bootstrap sample T' from T
Build a decision tree using T', but, at every split point:
- Choose a random fraction K of the remaining features,
- Pick the best feature (minimising cost) from that subset.
Add the tree to the set, *without pruning*
end for
return set of trees

For a test point **x**, get a response from each tree, and take a majority vote.

Random Forest | Vantaggi / Svantaggi

- Migliora le prestazioni di un singolo albero di decisione appreso su tutti i dati

- Perde la facile interpretabilità degli alberi e scala meno bene

- Aumenta in maniera contenuta la complessità temporale rispetto agli alberi, dato che ogni singolo albero è appreso solo su un sottoinsieme dei dati e a ogni split solo un sottoinsieme dei predittori viene preso in considerazione

- Può gestire dati anche di grande taglia prima di offrire rallentamenti evidenti

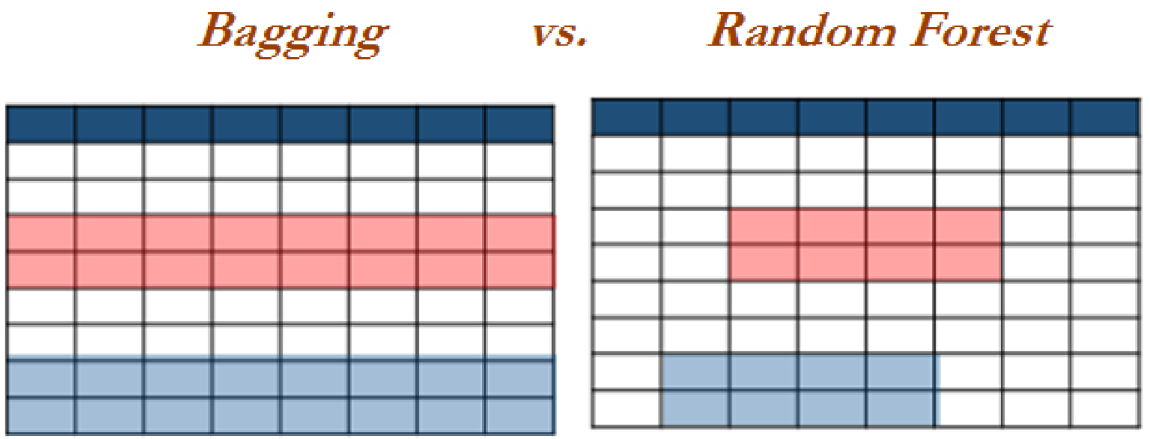
- Non presenta problemi di overfittng all'aumentare del numero di alberi. Questo è dovuto al fatto che solo una piccola porzione dei predittori è usata ad ogni split, oltre al fatto che RF predice mediante Aggregazione

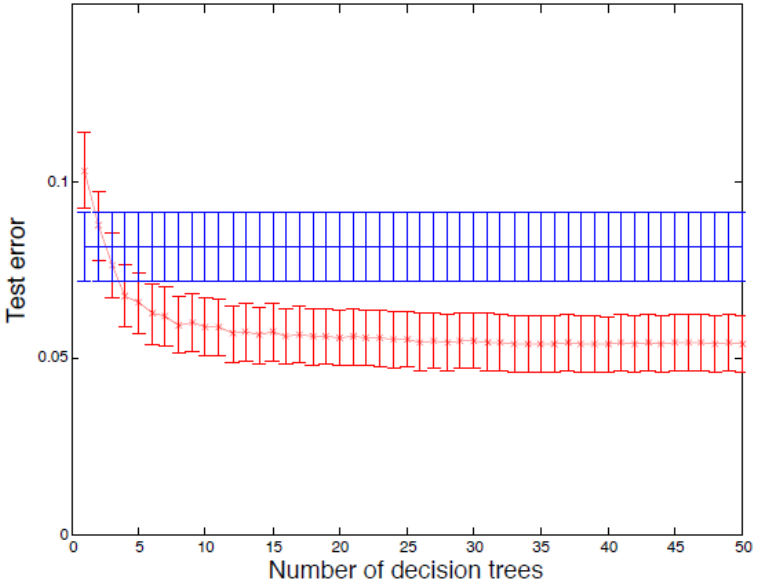
- È più stabile alle variazioni dei dati di input, grazie al bagging

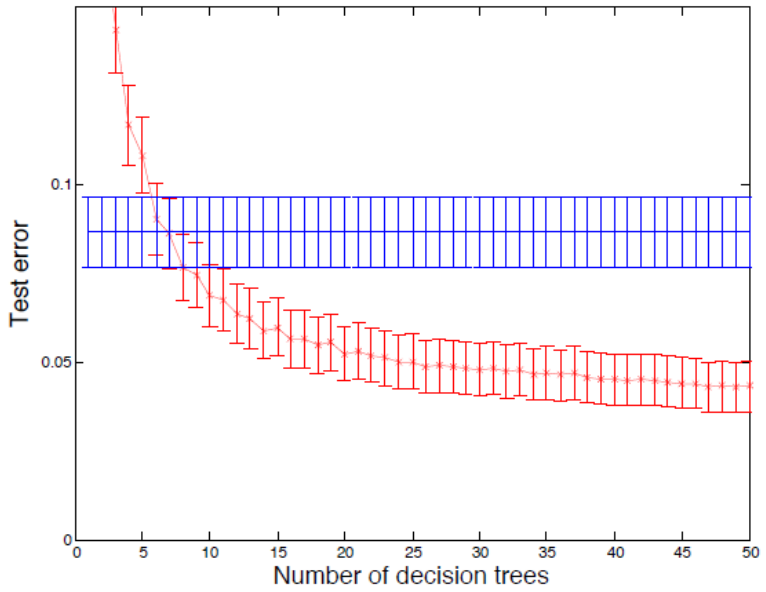
- Gli alberi appresi sono indipendenti

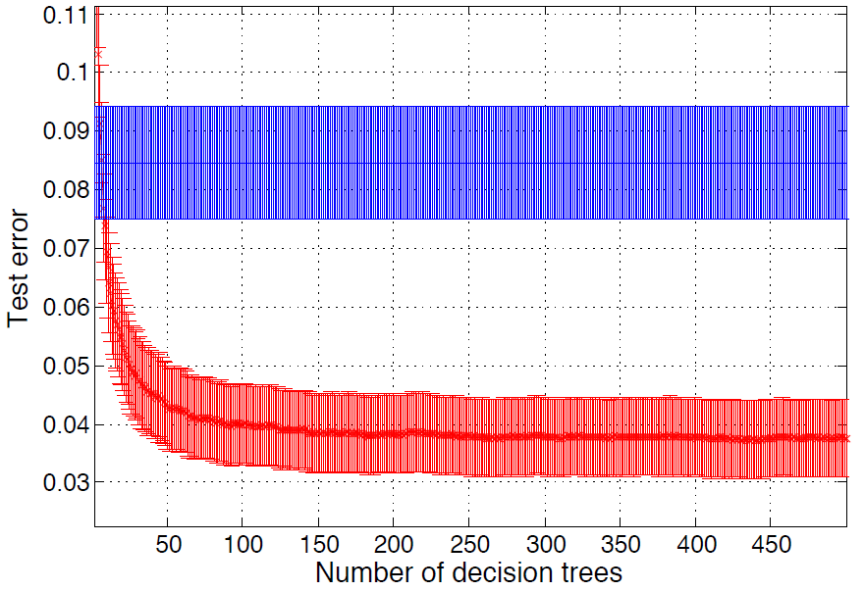
- Questo rende il metodo facilmente parallelizzabile con hardware appropriato (e.g. più core o processori)

Random Forest vs. (simple) Bagging





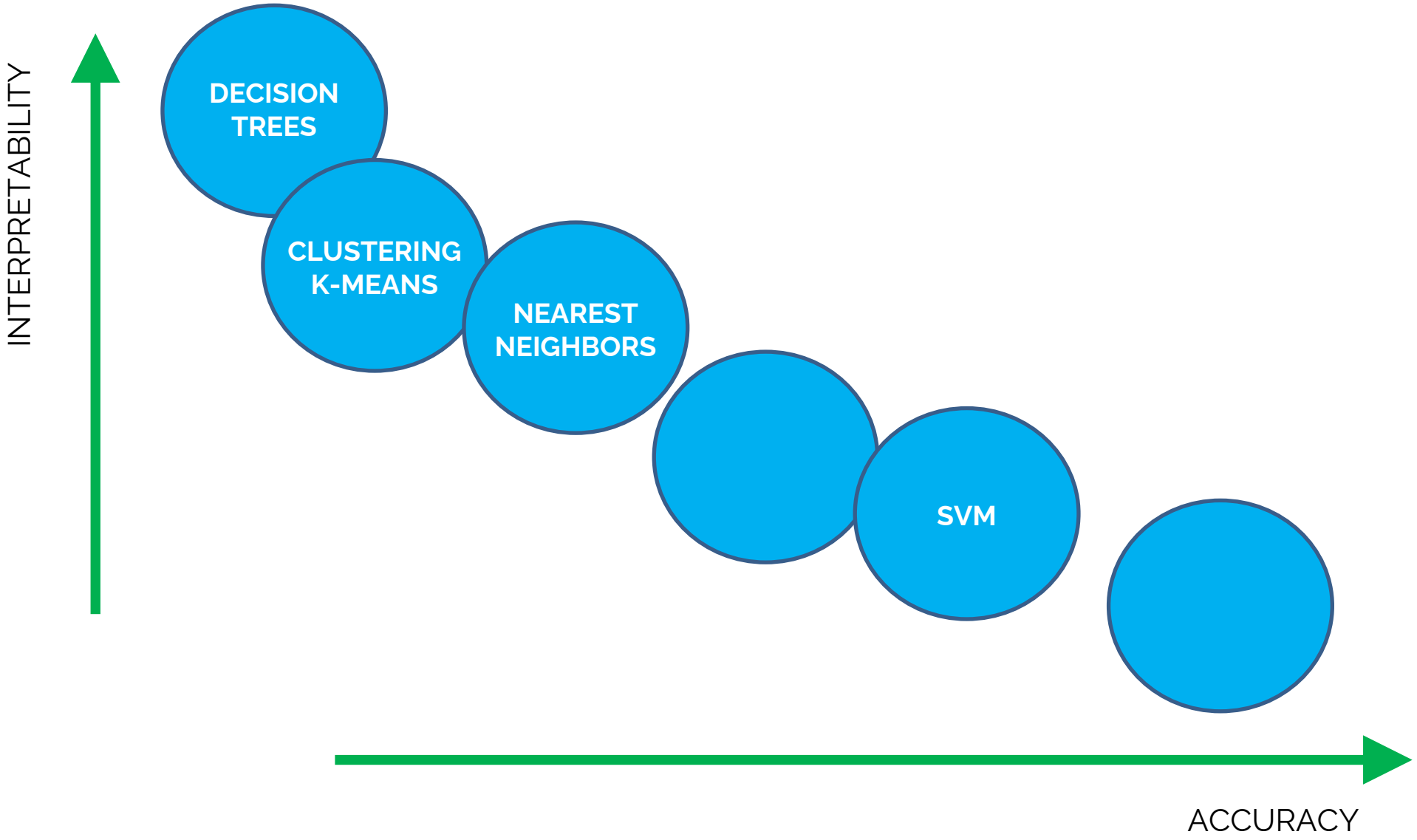




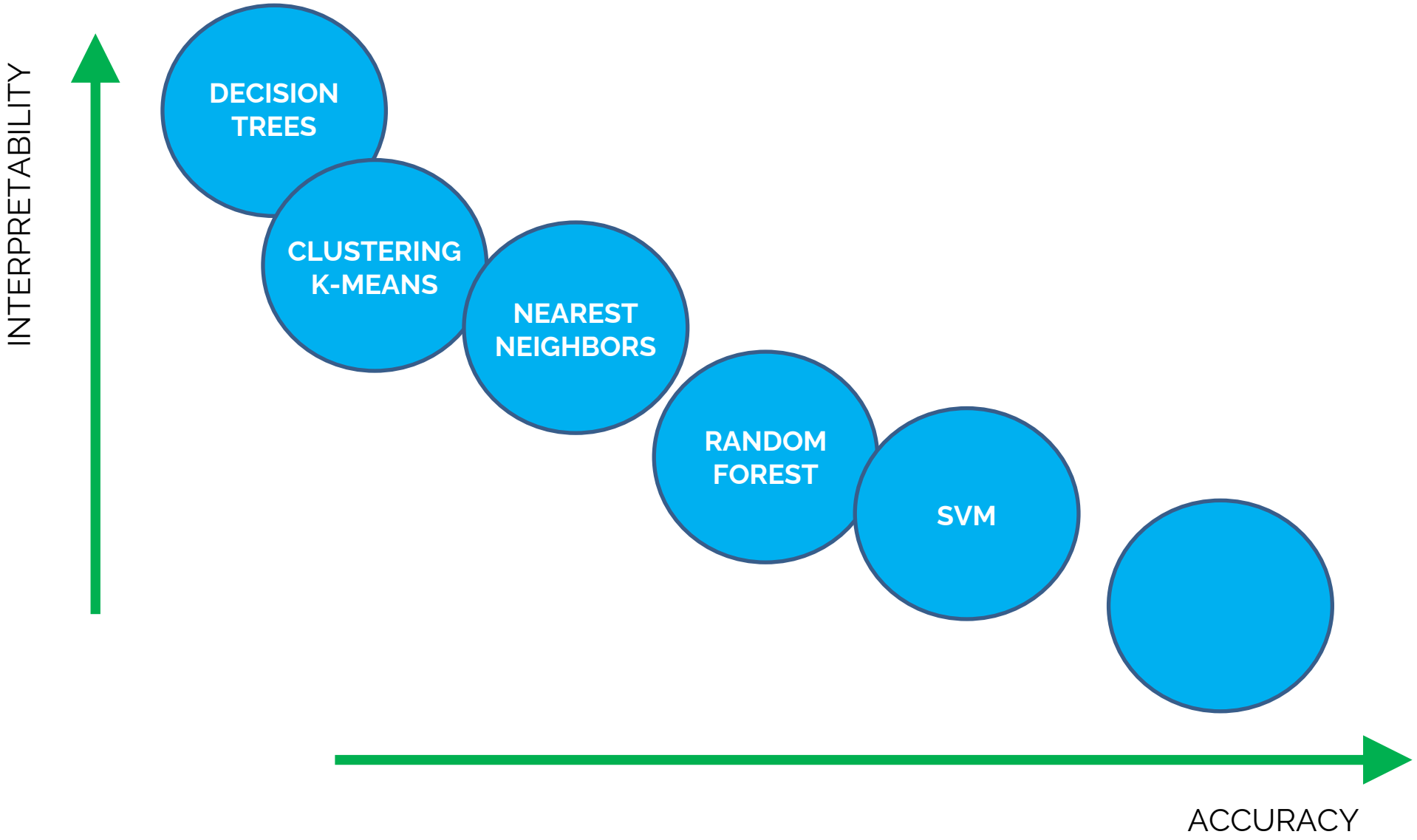
CREDITS: PROF. GAVIN BROWN

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

Interpretability-Accuracy TRADEOFF



Interpretability-Accuracy TRADEOFF



Which is which | Decision Function