Machine Learning 520
Advanced Machine Learning

Lesson 5: Stacking and Blending

UNIVERSITY of WASHINGTON

Week 3 Survey

Highlights

- The response rate is very high for ML.
- Students seem unanimously appreciated class structure and approach from multiple angles- demo, exercises, theory etc. which is great.
- Students feel the class is broken down well and class discussion is engaging.

To be Improved

- There are 4-5 comments to include more depth on the algorithms/math
- Length of the class. Leverage extra time to include mode depth.

Today's Agenda

- Combine classifiers
- Stacking
- Blending

Learning Objectives

By the end of this session, you should be able to:

- Describe the theory of how ensemble learning reduces errors.
- Use stacking to improve model performance.
- Use blending to improve model performance.

Recap

- Lesson 3 Assignment Solution
- SVM Recap

SVM Recap

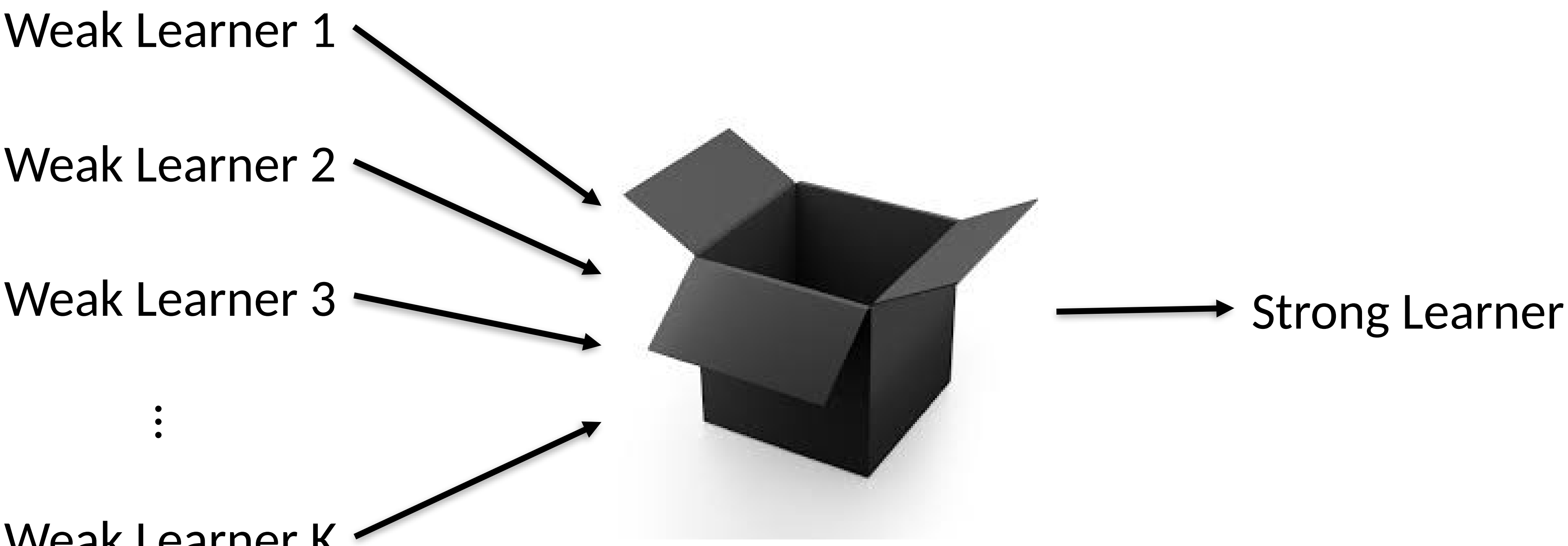
- Maximum Margin
- Hinge loss
- SVM with linear kernel
- Kernel tricks
  - SVM with polynomial kernel
  - SVM with radial basis function kernel
  - Support Vector Regression

Weak Learner & Strong Learner

- A **weak learner**: it can make predictions (slightly) **better than random guessing**.
  - Weak learners have high bias and cannot solve hard learning problems.
  - e.g., naïve Bayes, logistic regression, decision stumps (decision trees of depth 1)

- A **strong learner**: it has **arbitrarily small error rate**.
  - Strong learners are our goal of machine learning.
  - e.g., random forest, deep neural networks

Can we turn weak learners into a strong one?



Weak Learner 1

Weak Learner 2

Weak Learner 3

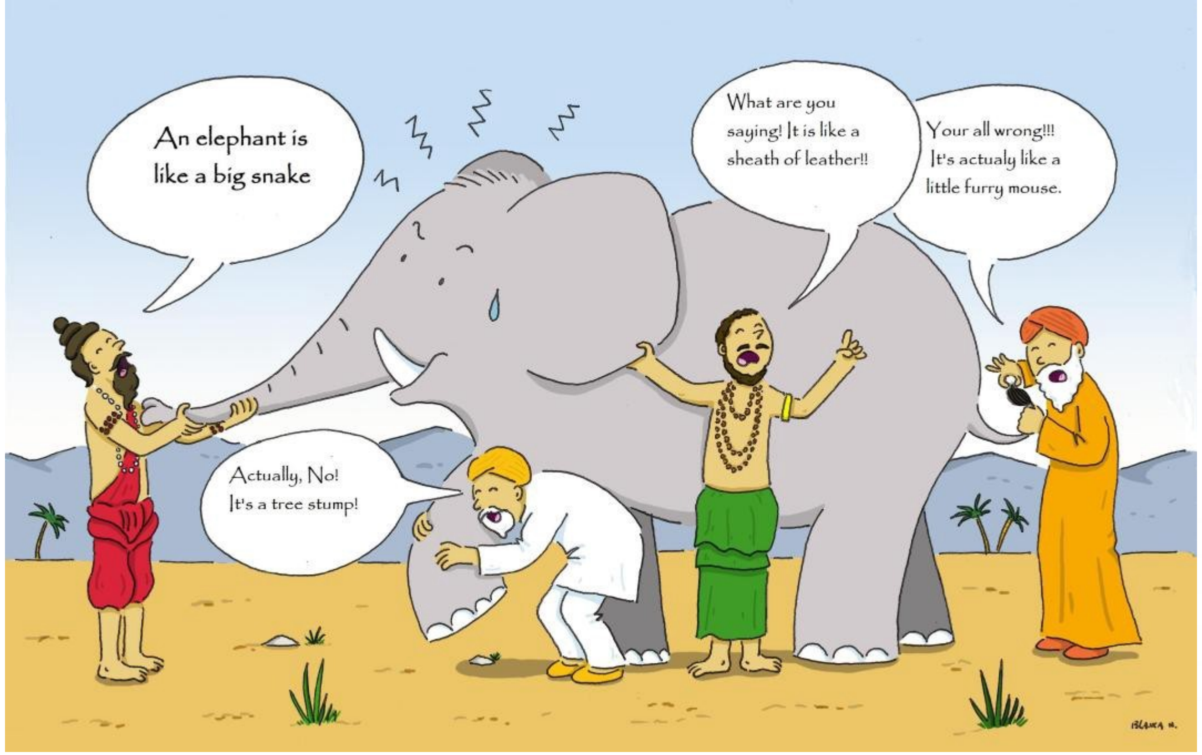
⋮

Weak Learner K

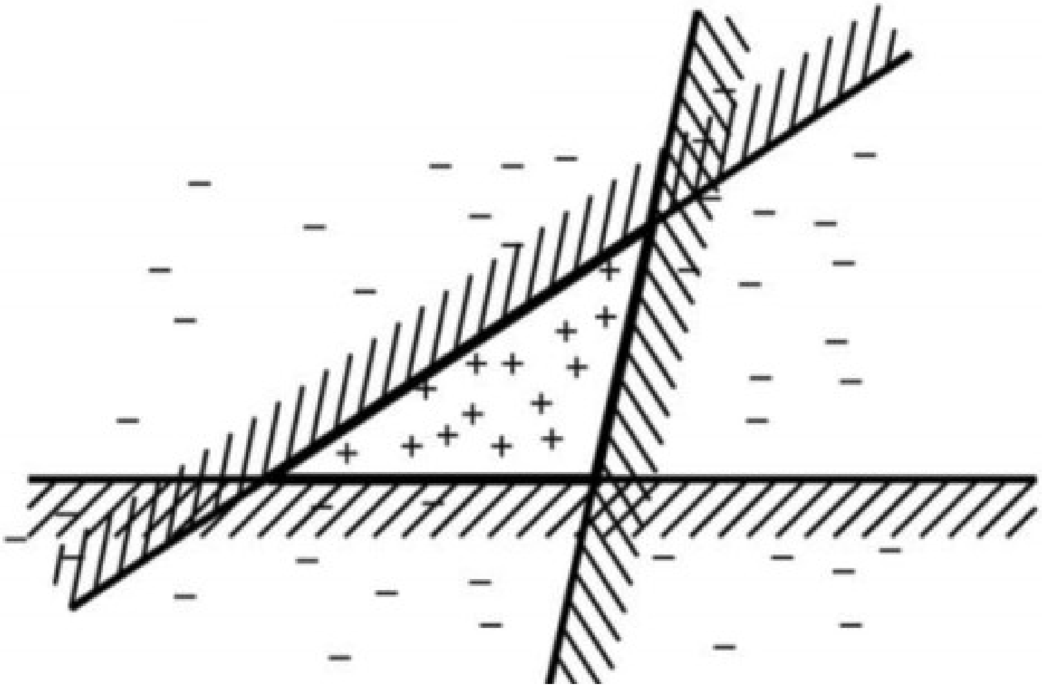
Strong Learner

YES, ENSEMBLE LEARNING

Ensemble Learning Intuition


An elephant is like a big snake
Actually, No! It's a tree stump!
What are you saying! It is like a sheath of leather!!
Your all wrong!!! It's actualy like a little furry mouse.

Fable of blind men and elephant



Combine 3 linear classifiers

Ensemble Learning

> Instead of learning a single classifier, we learn a set of classifiers.

How do we learn a set of classifiers?

> Combine the predictions of multiple classifiers to produce the final prediction.

How do we combine all the classifiers? **Can you give real-life example**

Bagging

Varies data set

Each training set a *bootstrap* sample

- bootstrap sample - select set of examples (with replacement) from original sample

Algorithm:

```
for $k$ = 1 to #classifiers
    train´ = bootstrap sample of train set
    create classifier using train´ as training set
combine classifications using simple voting
```

Weak Learning

Schapire showed that a set of weak learners (learners with > 50% accuracy, but not much greater) could be combined into a strong learner

Idea: weight the data set based on how well we have predicted data points so far

- data points predicted accurately - low weight
- data points mispredicted - high weight

Result: focuses components on portion of data space not previously well predicted

Boosting

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
  - Initially, all N records are assigned equal weights
  - Unlike bagging, weights may change at the end of a boosting round

Boosting

- Records that are wrongly classified will have their weights increased
- Records that are classified correctly will have their weights decreased

Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

- Example 4 is hard to classify
- Its weight is increased, therefore it is more likely to be chosen again in subsequent rounds

Boosting

- Equal weights are assigned to each training instance (1/N for round 1) at first round
- After a classifier C_i is learned, the weights are adjusted to allow the subsequent classifier

C_{i+1} to “pay more attention” to data that were misclassified by C_i.

- Final boosted classifier C^* combines the votes of each individual classifier
  - Weight of each classifier’s vote is a function of its accuracy
- Adaboost – popular boosting algorithm

Adaboost (Adaptive Boost)

- Input:
  - Training set D containing ***N*** instances
  - *T* rounds
  - A classification learning scheme
- Output:
  - A composite model

Boosting - Adaboost: Training Phase

- Training data D contain N labeled data (X_1,y_1), $(X_2,y_2\ )$, (X_3,y_3),….(X_N,y_N)
- Initially assign equal weight 1/d to each data
- To generate *T* base classifiers, we need *T* rounds or iterations
- Round i, data from D are sampled with replacement , to form Di (size *N*)
- Each data's chance of being selected in the next rounds depends on its weight
  - Each time the new sample is generated directly from the training data D with different sampling probability according to the weights; these weights are not zero

Boosting - Adaboost: Training Phase

- Base classifier C_i, is derived from training data of Di
- Error of C_i is tested using Di
- Weights of training data are adjusted depending on how they were classified
  - Correctly classified: Decrease weight
  - Incorrectly classified: Increase weight
- Weight of a data indicates how hard it is to classify it (directly proportional)

Boosting - Adaboost: Testing Phase

- The lower a classifier error rate, the more accurate it is, and therefore, the higher its weight for voting should be
- Weight of a classifier C_i's vote is

$$\alpha_i = \frac{1}{2}\ln\left(\frac{1 - \varepsilon_i}{\varepsilon_i}\right)$$

- Testing:
  - For each class c, sum the weights of each classifier that assigned class c to X (unseen data)
  - The class with the highest sum is the WINNER!

$$C^*(x_{test}) = \arg\max_{y} \sum_{i=1}^{T} \alpha_i \delta\left(C_i(x_{test}) = y\right)$$

Example: Error and Classifier Weight in AdaBoost

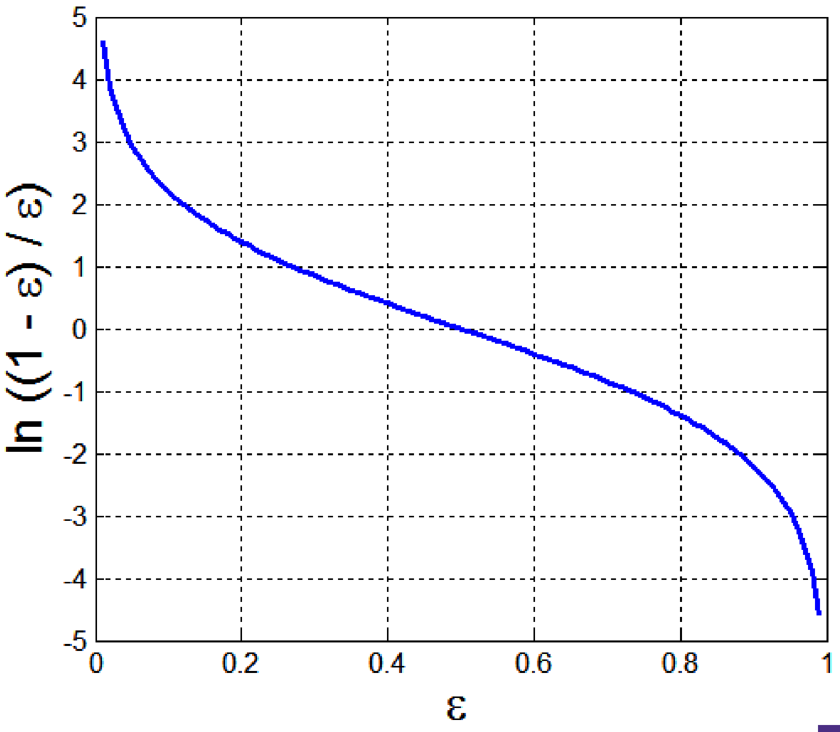
- Base classifiers: C_1, C_2, …, C_T

- Error rate: (*i* = index of classifier, *j*=index of instance)

$$\varepsilon_i = \frac{1}{N} \sum_{j=1}^{N} w_j \delta\left(C_i(x_j) \neq y_j\right)$$

- Importance of a classifier:

$$\alpha_i = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_i}{\varepsilon_i}\right)$$



Boosting – Arcing

Adapt[at]ive Resampling and Combining

Sample data set (like Bagging), but probability of data point being chosen weighted (like Boosting)

m_i = #number of mistakes made on point *i* by previous classifiers

probability of selecting point *i* :

$$prob_i = \frac{1 + m_i^{\,4}}{\sum_{j=0}^{N} 1 + m_j^{\,4}}$$

Value 4 chosen empirically

Combine using voting

Why do ensembles work?

- Suppose there are 25 classifiers where each classifier has an error rate of 0.35.
  - Assume classifiers are **independent**: a mistake from one classifier does not depend on the predictions from other classifiers.
  - In practice they are NOT completely independent.

- *Majority Voting*: The ensemble makes a wrong prediction if the majority of the classifiers predict the wrong prediction.

- What is the probability that the ensemble makes a wrong prediction? (hint: 13 or more classifiers make wrong predictions).

Why do ensembles work?

- Suppose there are 25 base classifiers
  - Each classifier has error rate, ε = 0.35
  - Assume classifiers are independent
  - Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^{i} (1 - \varepsilon)^{25-i} = 0.06$$

How it works

- Majority vote
- Suppose we have 5 completely independent classifiers…
  - If accuracy is 70% for each
    - $(.7^5)+5(.7^4)(.3)+ 10\ (.7^3)(.3^2)$
    - **83.7% majority vote accuracy**
  - 101 such classifiers
    - **99.9% majority vote accuracy**

Note: Binomial Distribution: The probability of observing *x* heads in a sample of *n* independent coin tosses, where in each toss the probability of heads is *p*, is

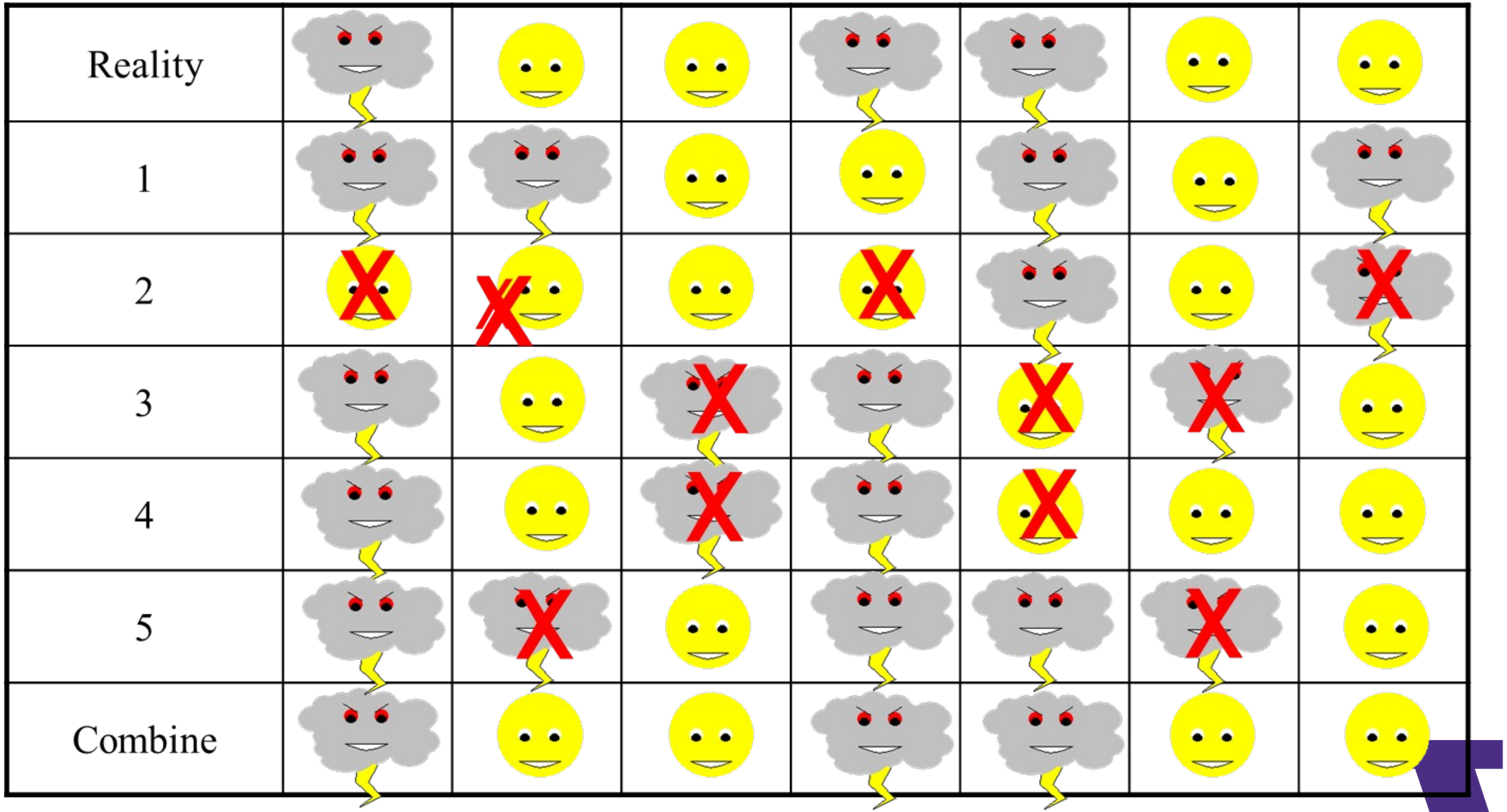
$$P(X = x|p, n) = \frac{n!}{r!(n-x)!}p^x(1-p)^{n-x}$$

Value of Ensembles

- “No Free Lunch” Theorem
  - No single algorithm wins all the time!

- When combing multiple independent and diverse decisions each of which is at least more accurate than random guessing, random errors cancel each other out, correct decisions are reinforced.

Example: Weather Forecast



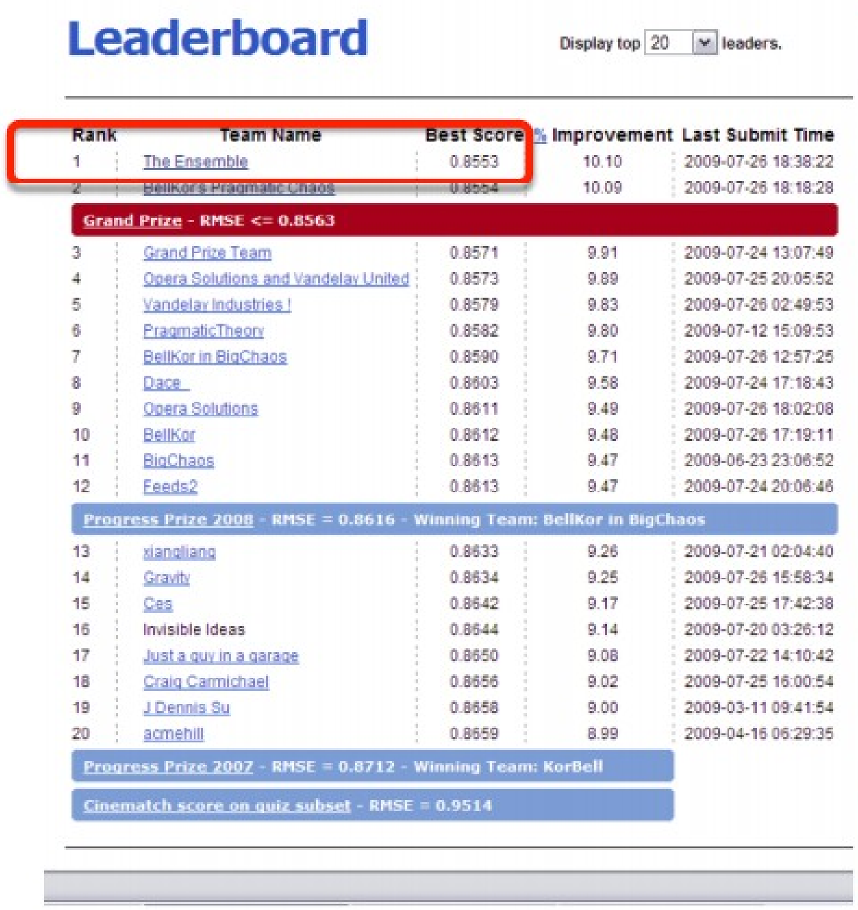
Reality	⛈	☀	☀	⛈	⛈	☀	☀
1	⛈	⛈	☀	☀	⛈	☀	⛈
2	☀ ✗	☀ ✗	☀	☀ ✗	⛈	☀	⛈ ✗
3	⛈	☀	⛈ ✗	⛈	☀ ✗	⛈ ✗	☀
4	⛈	☀	⛈ ✗	⛈	☀ ✗	☀	☀
5	⛈	⛈ ✗	☀	⛈	⛈	⛈ ✗	☀
Combine	⛈	☀	☀	⛈	⛈	☀	☀

Ensemble Mechanisms - Combiners

- Voting
- Averaging (if predictions not 0,1)
- Weighted Averaging
  - base weights on confidence in component
- Learning combiner
  - Stacking, Wolpert
    - general combiner
  - RegionBoosting, Maclin
    - piecewise combiner

Ensemble Learning in Netflix Prize

Machine learning competition with a $1 million prize



Leaderboard

Display top 20 leaders.

Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	The Ensemble	0.8553	10.10	2009-07-26 18:38:22
2	BellKor's Pragmatic Chaos	0.8554	10.09	2009-07-26 18:18:28
Grand Prize - RMSE <= 0.8563				
3	Grand Prize Team	0.8571	9.91	2009-07-24 13:07:49
4	Opera Solutions and Vandelay United	0.8573	9.89	2009-07-25 20:05:52
5	Vandelay Industries !	0.8579	9.83	2009-07-26 02:49:53
6	PragmaticTheory	0.8582	9.80	2009-07-12 15:09:53
7	BellKor in BigChaos	0.8590	9.71	2009-07-26 12:57:25
8	Dace_	0.8603	9.58	2009-07-24 17:18:43
9	Opera Solutions	0.8611	9.49	2009-07-26 18:02:08
10	BellKor	0.8612	9.48	2009-07-26 17:19:11
11	BigChaos	0.8613	9.47	2009-06-23 23:06:52
12	Feeds2	0.8613	9.47	2009-07-24 20:06:46
Progress Prize 2008 - RMSE = 0.8616 - Winning Team: BellKor in BigChaos				
13	xiangliang	0.8633	9.26	2009-07-21 02:04:40
14	Gravity	0.8634	9.25	2009-07-26 15:58:34
15	Ces	0.8642	9.17	2009-07-25 17:42:38
16	Invisible Ideas	0.8644	9.14	2009-07-20 03:26:12
17	Just a guy in a garage	0.8650	9.08	2009-07-22 14:10:42
18	Craig Carmichael	0.8656	9.02	2009-07-25 16:00:54
19	J Dennis Su	0.8658	9.00	2009-03-11 09:41:54
20	acmehill	0.8659	8.99	2009-04-16 06:29:35
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell				
Cinematch score on quiz subset - RMSE = 0.9514				

NETFLIX
Browse
Recommendations
Friends
Queue
Buy DVDs
Home
Genres
New Releases
Previews
Netflix Top 100
Movies For You
You really liked it...
The Big One

movies	users					
	1		?	3	5	?
	?	1				2
		4		4	5	?

The Ensemble was an ensemble solution of teams which had been competing individually for the prize.

Quiz

Which of the following algorithm is not an example of an ensemble method?

A. Extra Tree Regressor
B. Random Forest
C. Gradient Boosting
D. Decision Tree ←

Quiz

What is true about an ensembled classifier?

1. Classifiers that are more “sure” can vote with more conviction
2. Classifiers can be more “sure” about a particular part of the space
3. Most of the times, it performs better than a single classifier

A. 1 and 2
B. 1 and 3
C. 2 and 3
D. All of the above

Quiz

Refer below table for models M1, M2 and M3.

Actual Output	M1	M2	M3	Output
1	1	0	0	
1	1	1	1	
1	1	0	0	
1	0	1	0	
1	0	1	1	
1	0	0	1	
1	1	1	1	
1	1	1	1	
1	1	1	1	
1	1	1	1	

If you want to ensemble these models using majority voting. What will be the minimum accuracy you can get?

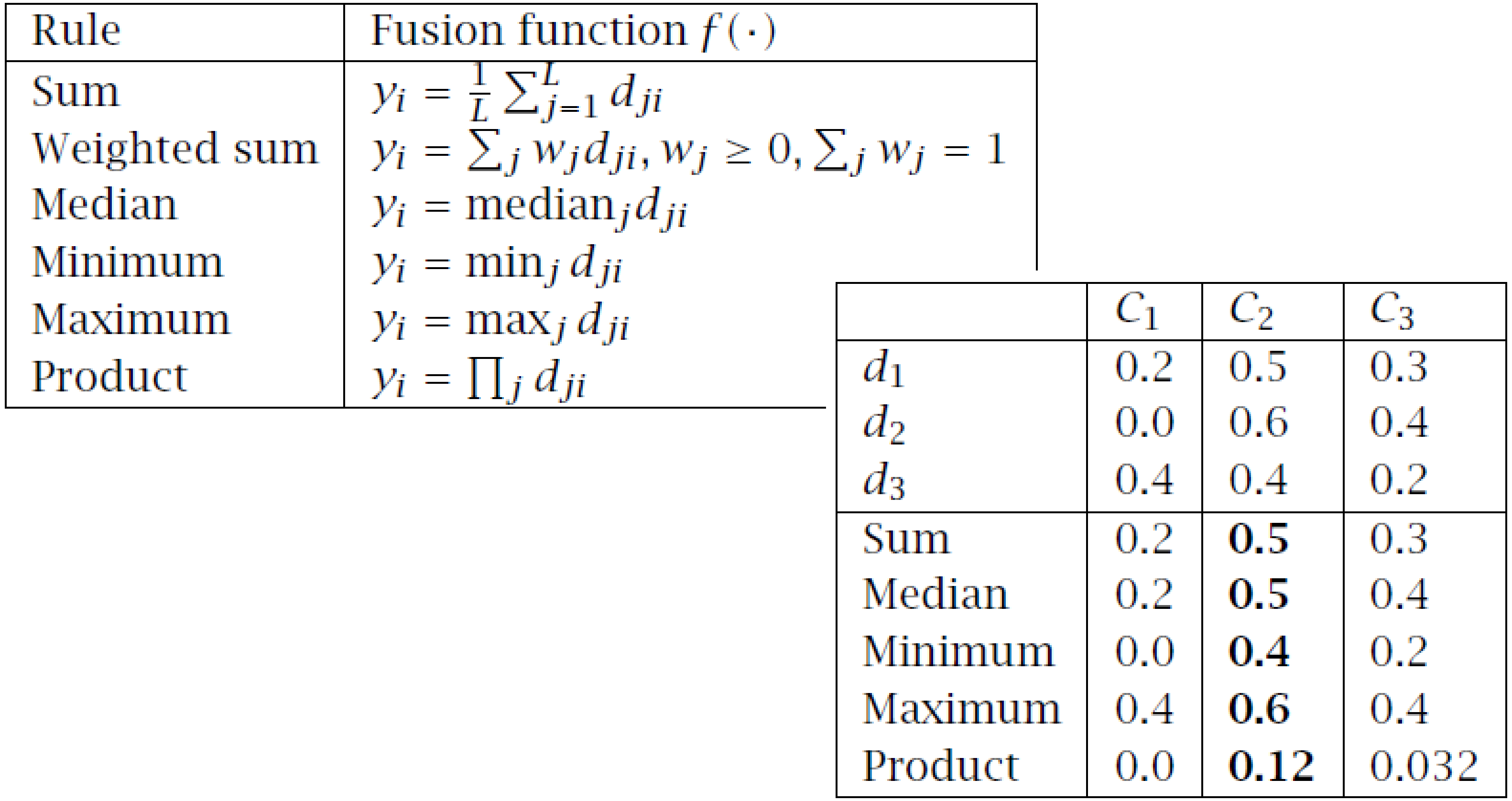
A. Always greater than 70%

B. Always greater than and equal to 70%

[✓ C. It can be less than 70%]

D. None of these

Fixed Combination Rules



Rule	Fusion function $f(\cdot)$
Sum	$y_i = \frac{1}{L}\sum_{j=1}^{L} d_{ji}$
Weighted sum	$y_i = \sum_j w_j d_{ji}, w_j \geq 0, \sum_j w_j = 1$
Median	$y_i = \text{median}_j d_{ji}$
Minimum	$y_i = \min_j d_{ji}$
Maximum	$y_i = \max_j d_{ji}$
Product	$y_i = \prod_j d_{ji}$

	C_1	C_2	C_3
d_1	0.2	0.5	0.3
d_2	0.0	0.6	0.4
d_3	0.4	0.4	0.2
Sum	0.2	**0.5**	0.3
Median	0.2	**0.5**	0.4
Minimum	0.0	**0.4**	0.2
Maximum	0.4	**0.6**	0.4
Product	0.0	**0.12**	0.032

Stacking

- Stacking or Stacked Generalization is an ensemble machine learning algorithm.

- It uses a meta-learning algorithm to learn how to best combine the predictions from two or more base machine learning algorithms.

- The benefit of stacking is that it can harness the capabilities of a range of well-performing models on a classification or regression task and make predictions that have better performance than any single model in the ensemble.

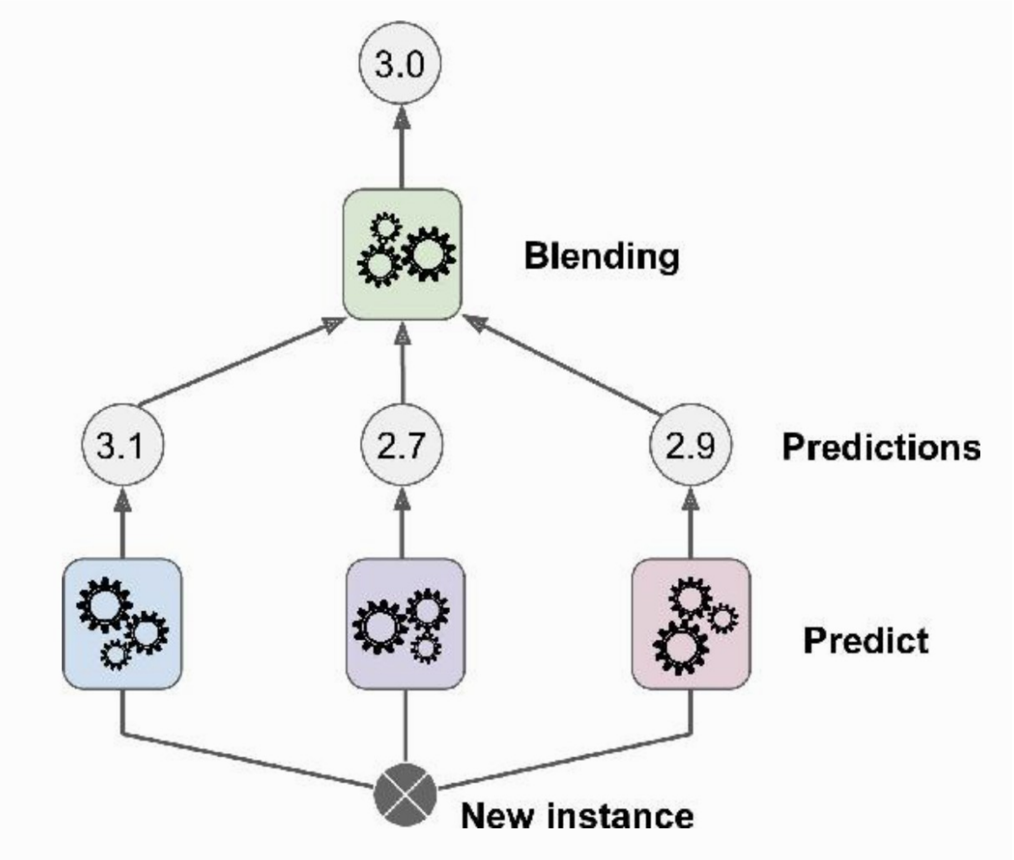
Blending

- Blending is an ensemble machine learning technique that uses a machine learning model to learn how to best combine the predictions from multiple contributing ensemble member models.
- As such, blending is the same as stacked generalization, known as stacking, broadly conceived. Often, blending and stacking are used interchangeably in the same paper or model description.

> *Many machine learning practitioners have had success using stacking and related techniques to boost prediction accuracy beyond the level obtained by any of the individual models. In some contexts, stacking is also referred to as blending, and we will use the terms interchangeably here.*
>
> — Feature-Weighted Linear Stacking, 2009.

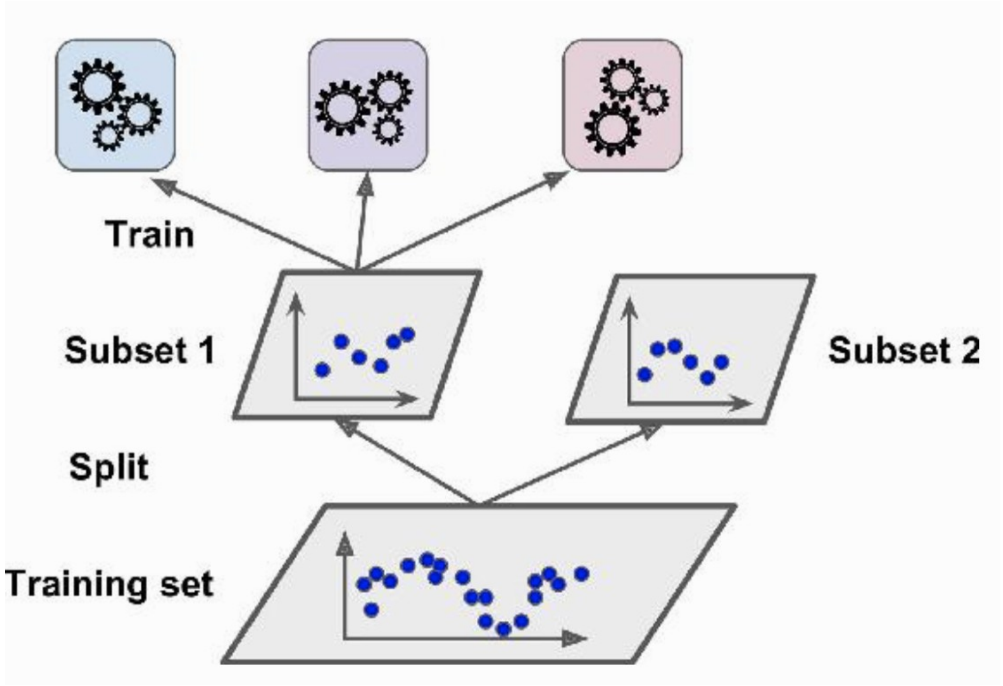
The Idea of Stacking

- Instead of using trivial functions (such as majority voting / averaging) to aggregate the predictions of all predictors in an ensemble, we **train a model (aka blender)** to perform this aggregation.



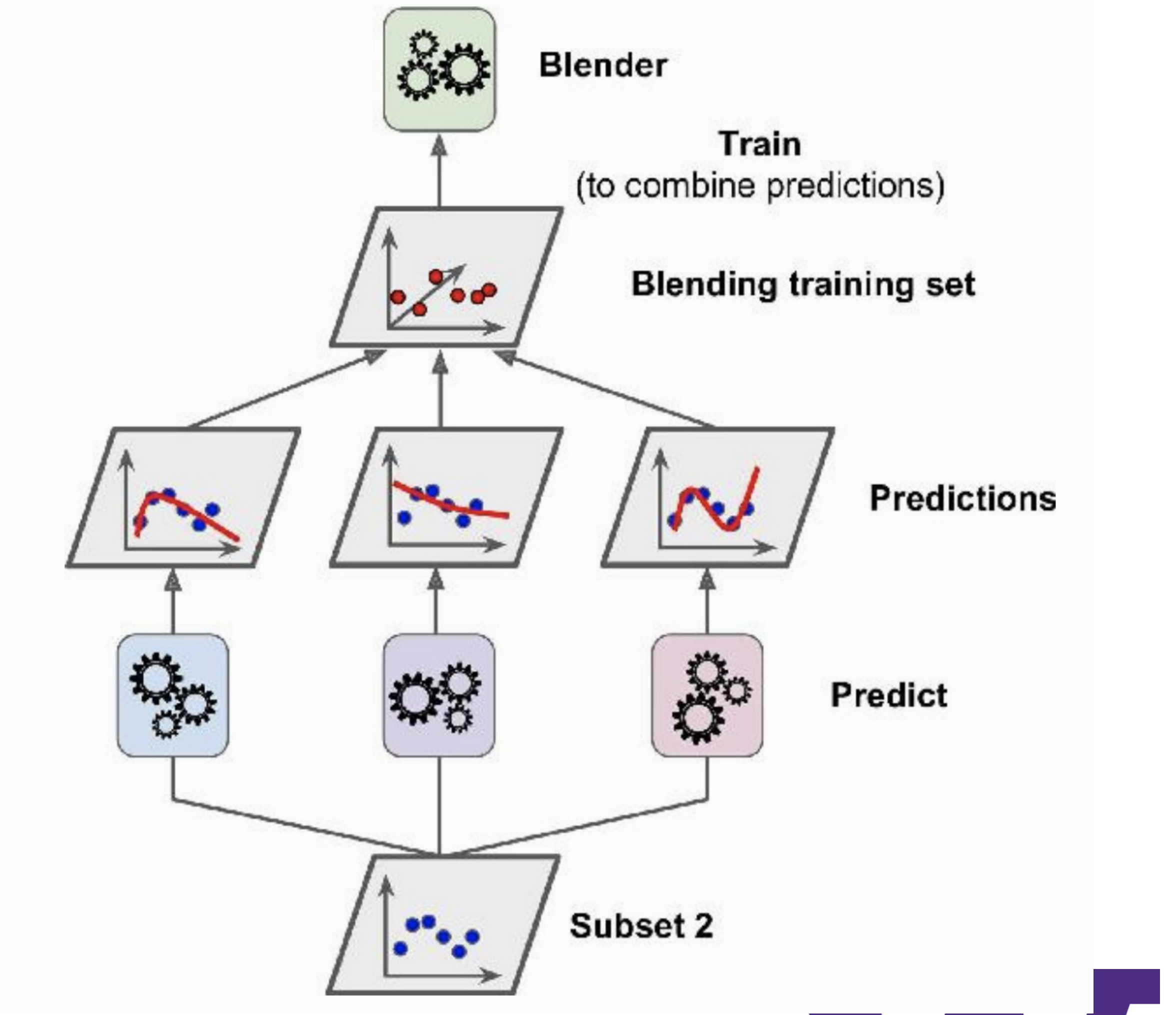
Training stacked model – Step 1

- The training set is split in two subsets.
- The first subset is used to train the predictors in the first layer.
- The second subset is used to train the blender in the second layer.



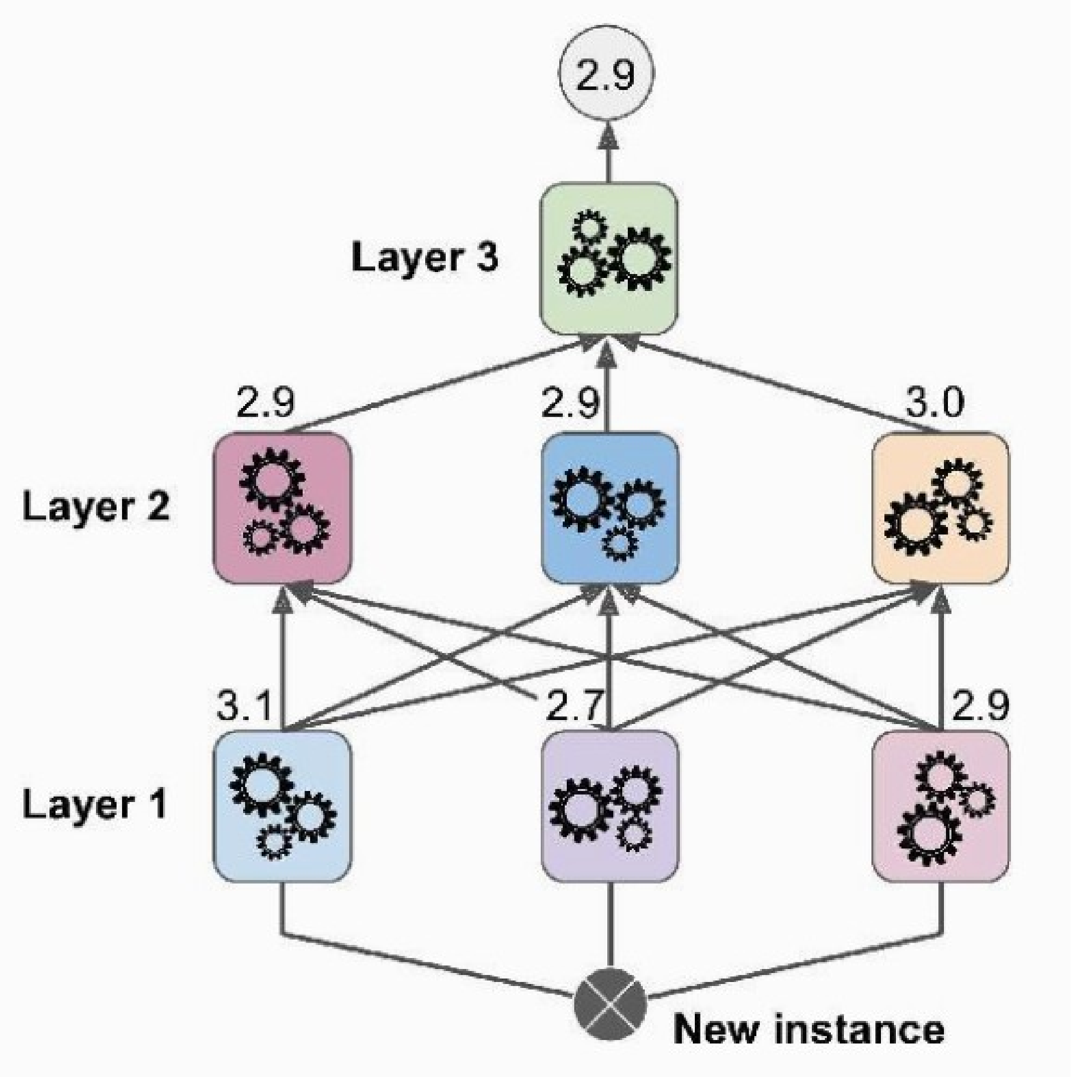
Training stacked model – Step 2

- The first layer predictors are used to make predictions on the second subset.
- Create a new training set using these predicted values as input features.
- The blender is trained on this new training set, so it learns to predict the target value given the first layer’s predictions.



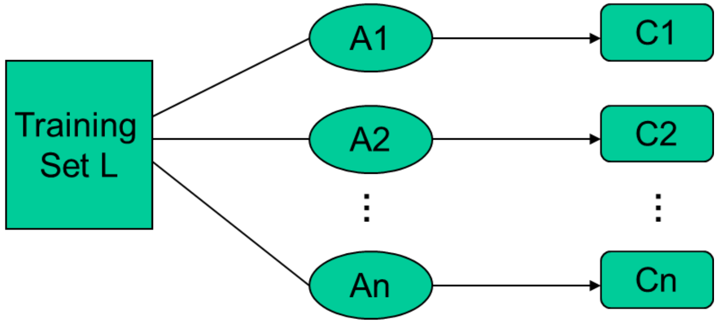
Multilayer Stacking Ensemble

- We can easily extend the 2-layer stacking model to multi-layer stacking.
- Split the original training data into K subsets for a K-layer stacking model.
- The ith subset of the data is used to learn the blenders in the ith layer to avoid data leakage.

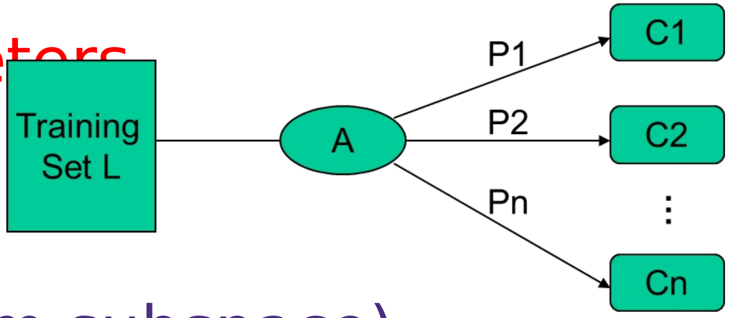


Types of Ensemble Learning

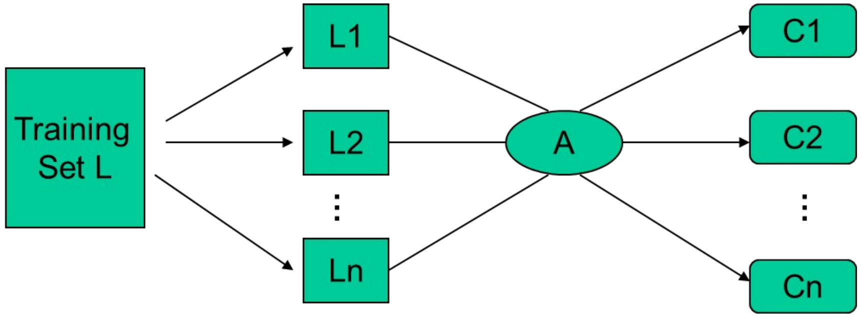
- Different learning algorithms



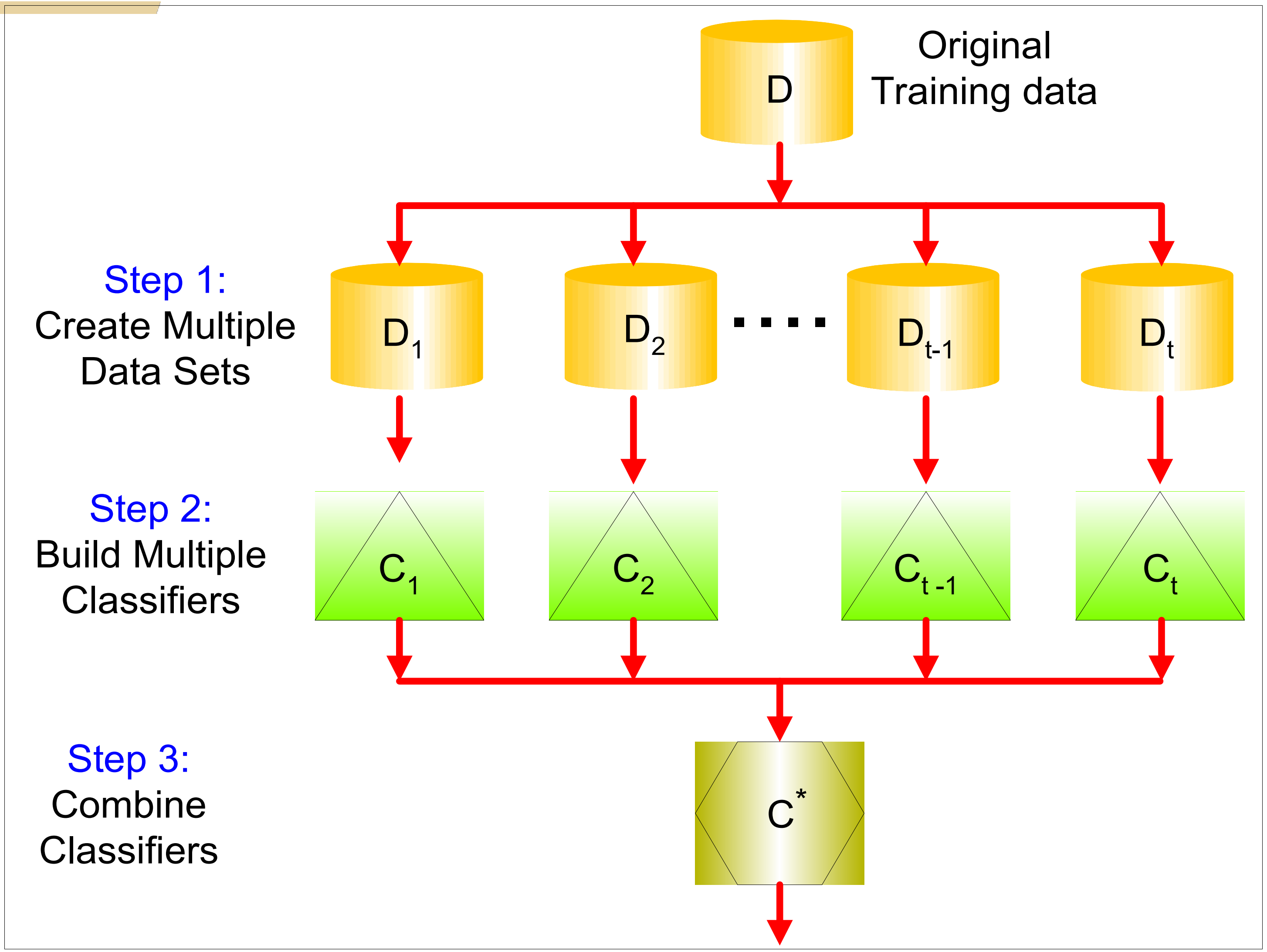
- Algorithms with different choice for parameters



- Data set with different features (e.g. random subspace)
- Data set = different subsets (e.g. bagging, boosting)



General Idea


D
Original Training data
Step 1: Create Multiple Data Sets
D_1
D_2
....
D_{t-1}
D_t
Step 2: Build Multiple Classifiers
C_1
C_2
C_{t-1}
C_t
Step 3: Combine Classifiers
C^*

How to Combine Classifiers

- Voting:
  - Classifiers are combined in a static way
  - Each base-level classifier gives a (weighted) vote for its prediction
  - Plurality vote: each base-classifier predict a class

Example

- Stacking: a stack of classifiers
  - Classifiers are combined in a dynamically
  - A machine learning method is used to learn how to combine the prediction of the base-level classifiers.
  - Top level classifier is used to obtain the final prediction from the predictions of the base-level classifiers

Example

Notebook Time

What is the Main Challenge for Developing Ensemble Models?

- The main challenge is not to obtain highly accurate base models, but rather to obtain base models which make different kinds of errors.
- For example, if ensembles are used for classification, high accuracies can be accomplished if different base models misclassify different training examples, even if the base classifier accuracy is low. Independence between two base classifiers can be assessed in this case by measuring the degree of overlap in training examples they misclassify (|A ∩B|/|A∪B|)—more overlap means less independence between two models.