COSC 522 – Machine Learning

Classifier Fusion

Hairong Qi, Gonzalez Family Professor
Electrical Engineering and Computer Science
University of Tennessee, Knoxville
https://www.eecs.utk.edu/people/hairong-qi/
Email: hqi@utk.edu

Roadmap

- Module 1: Baysian Decision Theory (Maximum Posterior Probability or MPP)
  - Parametric
  - Non-parametric
  - In-depth: the three cases – parametric (e.g., pdf is Gaussian)
    - Minimum Euclidean Distance Classifier – linear machine (features are independent, covariance matrices from different classes are the same, pdf is Gaussian)
    - Minimum Mahalanobis Distance Classifier – linear machine (~~features are independent~~, covariance matrices from different classes are the same, pdf is Gaussian)
    - Quadratic Machine (~~features are independent~~, ~~covariance matrices from different classes are the same~~, pdf is Gaussian)

$$P(w_j|x) = \frac{p(x|w_j)P(w_j)}{p(x)}$$

- Module 2: Connection-based Neural Networks
  - Perceptron
  - BPNN and MLP (multi-layer perceptron)

Test 1

  - Kernel Methods
  - SVM
- Module 3: Regression
  - Linear Regression
  - Logistic Regression
- Module 4: Unsupervised Learning
  - Assume k is known (k-means, wta)
  - Hierarchical methods (Agglomerative clustering)
- Module 5: Pre-processing: Dimensionality Reduction
  - Supervised (FLD)
  - Unsupervised (PCA)
- Module 6: Post-processing
  - Performance Evaluation
  - Fusion

Questions

- Rationale with fusion?
- Different flavors of fusion?
- The fusion hierarchy

- What is the cost function for Naïve Bayes?
- What is the procedure for Naïve Bayes?
- What is the limitation of Naïve Bayes?
- What is the procedure of Behavior-Knowledge-Space (BKS)?
- How does it resolve issues with NB?

- What is Boosting and what is its difference to committee-based fusion approaches?
- What is AdaBoost?

Motivation

Three heads are better than one.

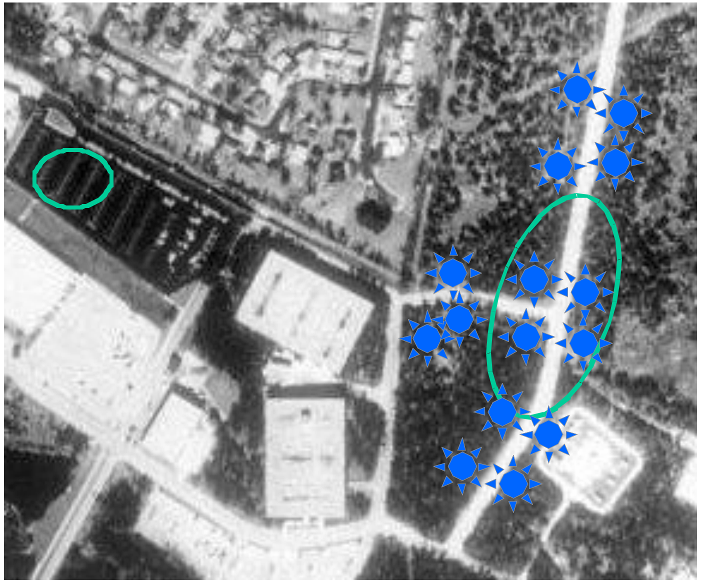
- Combining classifiers to achieve higher accuracy
  - Combination of multiple classifiers
  - Classifier fusion
  - Mixture of experts
  - Committees of neural networks
  - Consensus aggregation
  - …
- Reference:
  - L. I. Kuncheva, J. C. Bezdek, R. P. W. Duin, "Decision templates for multiple classifier fusion: an experimental comparison," *Pattern Recognition*, 34: 299-314, 2001.
  - Y. S. Huang and C. Y. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," IEEE Trans. Pattern Anal. Mach. Intell., vol. 17, no. 1, pp. 90–94, Jan. 1995.

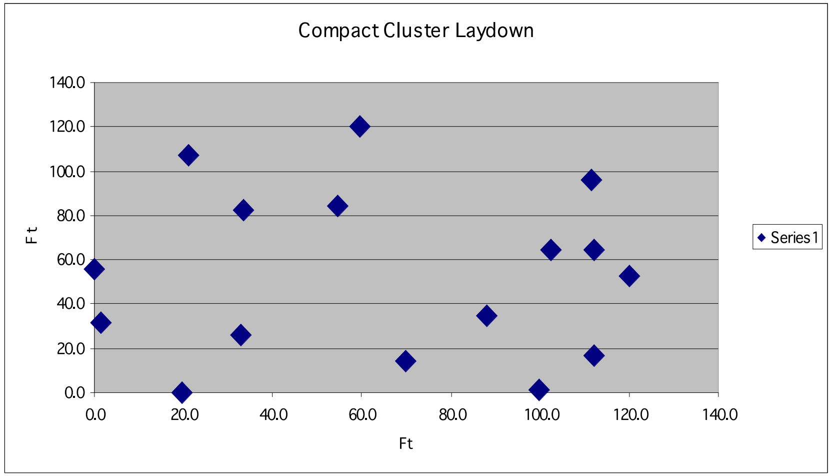
Popular Approaches

- Data-based fusion (early fusion)
- Feature-based fusion (middle fusion)
- Decision-based fusion (late fusion)

- Approaches
  - Committee-based
    - Majority voting
    - Bootstrap aggregation (Bagging) [Breiman, 1996]
  - Baysian-based
    - Naïve Bayes combination (NB)
    - Behavior-knowledge space (BKS) [Huang and Suen, 1995]
  - Boosting
    - Adaptive boosting (AdaBoost) [Freund and Schapire, 1996]
  - Interval-based integration

Application Example – Civilian Target Recognition





Ford 250

Harley Motocycle

Ford 350

Suzuki Vitara

Consensus Patterns

- Unanimity (100%)
- Simple majority (50%+1)
- Plurality (most votes)

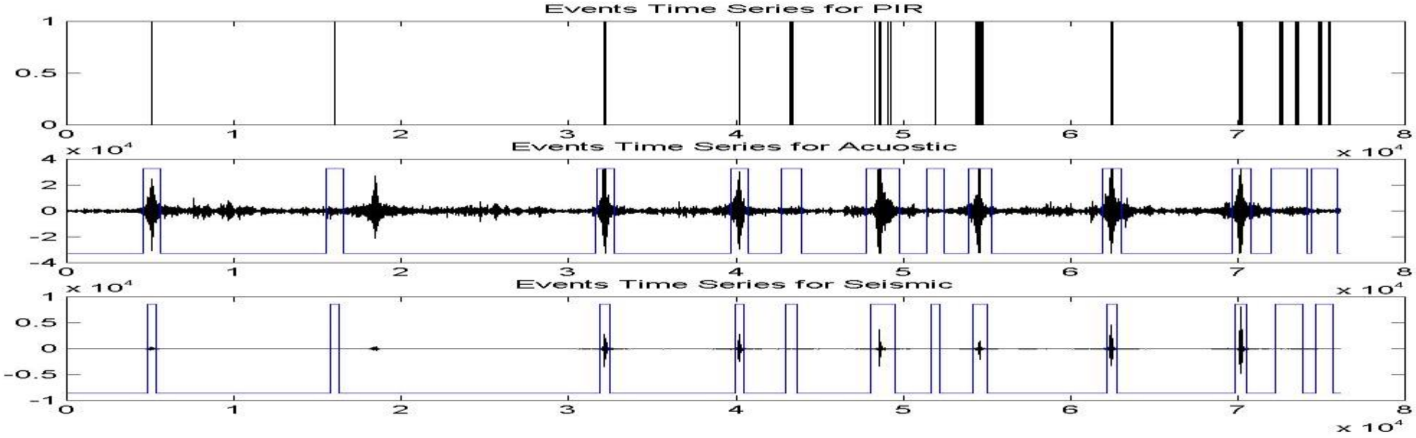
Example of Majority Voting - Temporal Fusion

- Fuse all the 1-sec sub-interval local processing results corresponding to the same event (usually lasts about 10-sec)
- Majority voting

$$\overline{j\,{}_i^j} = \arg\max w_c, \quad c\ \hat{\mathrm{I}}\ [1,C]$$

number of local output c occurrence

number of possible local processing results



PART I: BAYSIAN-BASED APPROACH

Naïve Bayes (the independence assumption)

AICIP RESEARCH

Confusion matrix

The real class is DW, the classifier says it’ s HMV

s

k

C1	AAV	DW	HMV
AAV	894	329	143
DW	99	411	274
HMV	98	42	713

C2	AAV	DW	HMV
AAV	1304	156	77
DW	114	437	83
HMV	13	107	450

i = 1, 2 (classifiers)

L1	AAV	DW	HMV
AAV			
DW			
HMV			

L2	AAV	DW	HMV
AAV			
DW			
HMV			

Probability that the true class is k given that C_i assigns it to s

Probability multiplication

NB – Derivation

- Assume the classifiers are mutually independent
- Bayes combination - Naïve Bayes, simple Bayes, idiot's Bayes
- Assume
  - L classifiers, i=1,..,L
  - c classes, k=1,…,c
  - s_i: class label given by the i^{th} classifier, i=1,…,L, $s=\{s_1,\ldots,s_L\}$

$$P(\omega_k|\mathbf{s}) = \frac{p(\mathbf{s}|\omega_k)P(\omega_k)}{p(\mathbf{s})} = \frac{P(\omega_k)\prod_{i=1}^{L} p(s_i|\omega_k)}{p(\mathbf{s})}$$

$$P(\omega_k) = N_k/N$$

$$p(s_i|\omega_k) = cm_{k,s_i}/N_k$$

$$P(\omega_k|\mathbf{s}) \approx \frac{1}{N_k^{L-1}} \prod_{i=1}^{L} cm_{k,s_i}$$

BKS

- Majority voting won’t work
- Behavior-Knowledge Space algorithm (Huang&Suen)

Assumption:

- 2 classifiers
- 3 classes
- 100 samples in the training set

Then:

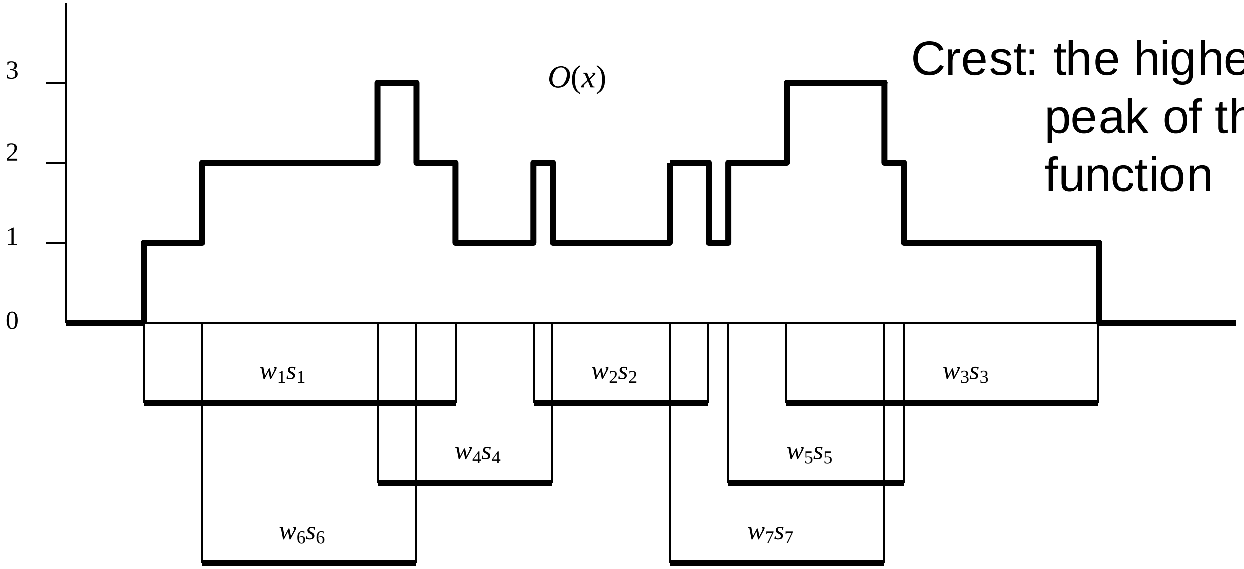
- 9 possible classification combinations

c_1, c_2	samples from each class	fused result
1,1	10/3/3	1
1,2	3/0/6	3
1,3	5/4/5	1,3
	…	
3,3	0/0/6	3

PART II: INTERVAL-BASED APPROACH

Value-based vs. Interval-based Fusion

- Interval-based fusion can provide fault tolerance
- Interval integration – overlap function
  - Assume each sensor in a cluster measures the same parameters, the integration algorithm is to construct a simple function (overlap function) from the outputs of the sensors in a cluster and can resolve it at different resolutions as required



Crest: the highest, widest peak of the overlap function

A Variant of kNN

- Generation of local confidence ranges (For example, at each node i, use kNN for each k∈{5,…,15})

	Class 1	Class 2	…	Class n
k=5	3/5	2/5	…	0
k=6	2/6	3/6	…	1/6
…	…	…	…	…
k=15	10/15	4/15	…	1/15
	{2/6, 10/15}	**{4/15, 3/6}**	…	**{0, 1/6}**

confidence level

confidence range

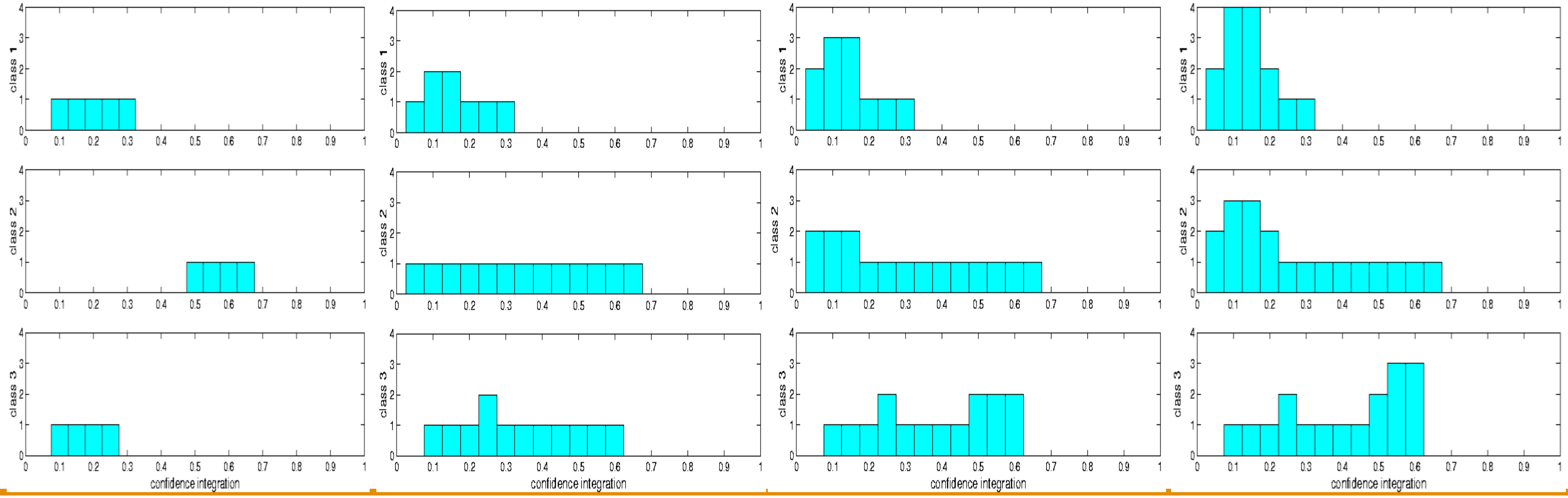
smallest

largest in this column

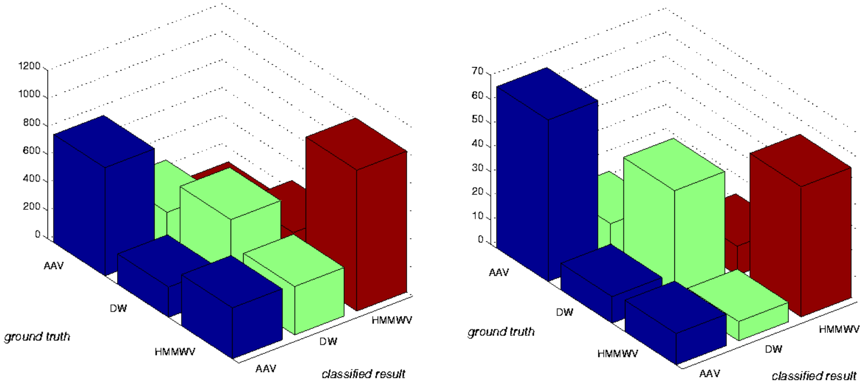
- Apply the integration algorithm on the confidence ranges generated from each node to construct an overlapping function

Example of Interval-based Fusion

	stop 1		stop 2		stop 3		stop 4	
	c	acc	c	acc	c	acc	c	acc
class 1	1	0.2	0.5	0.125	0.75	0.125	1	0.125
class 2	2.3	0.575	4.55	0.35	0.6	0.1	0.75	0.125
class 3	0.7	0.175	0.5	0.25	3.3	0.55	3.45	0.575

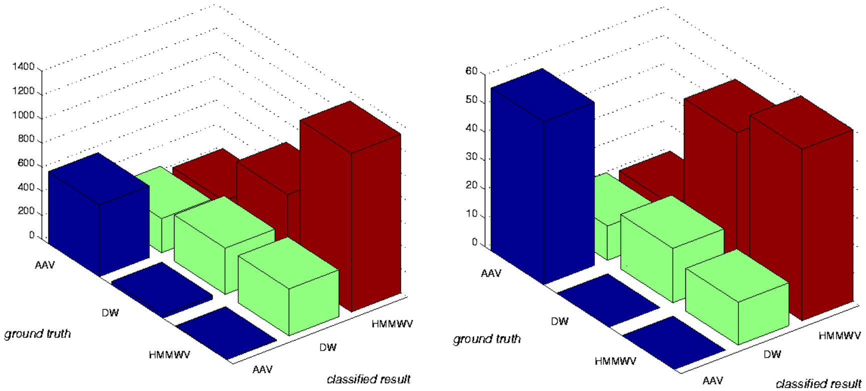


Confusion Matrices of Classification on Military Targets

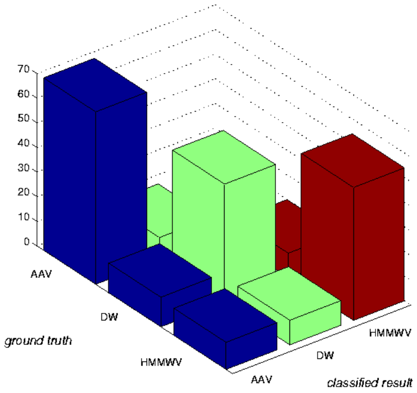


Acoustic (75.47%, 81.78%)

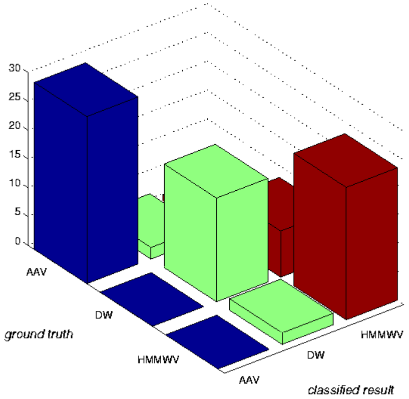
	AAV	DW	HMV
AAV	29	2	1
DW	0	18	8
HMV	0	2	23



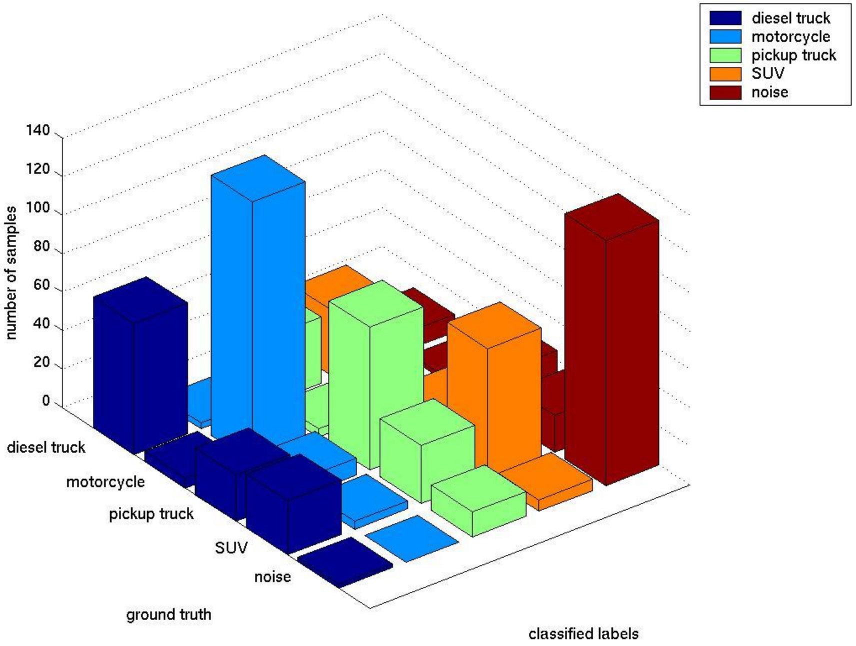
Seismic (85.37%, 89.44%)



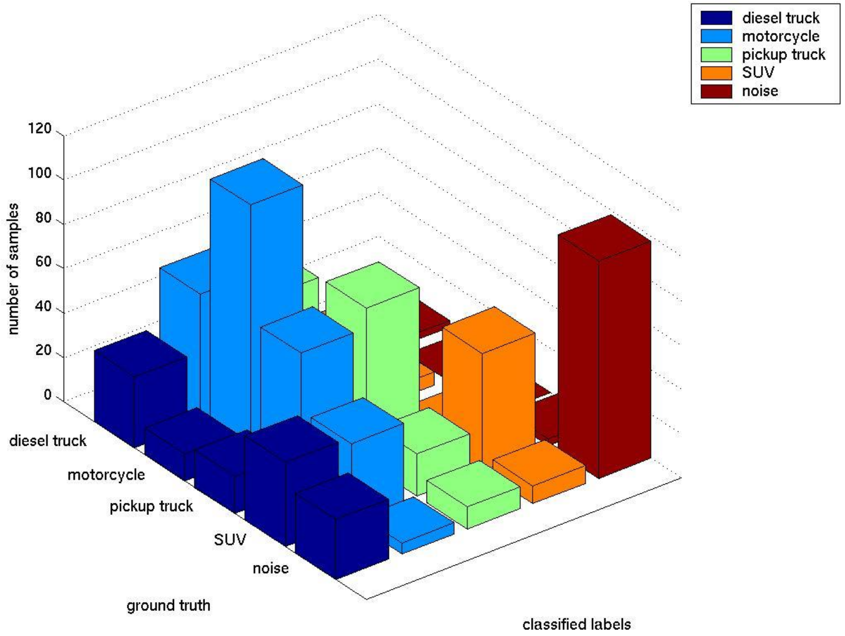
Multi-modality fusion (84.34%)



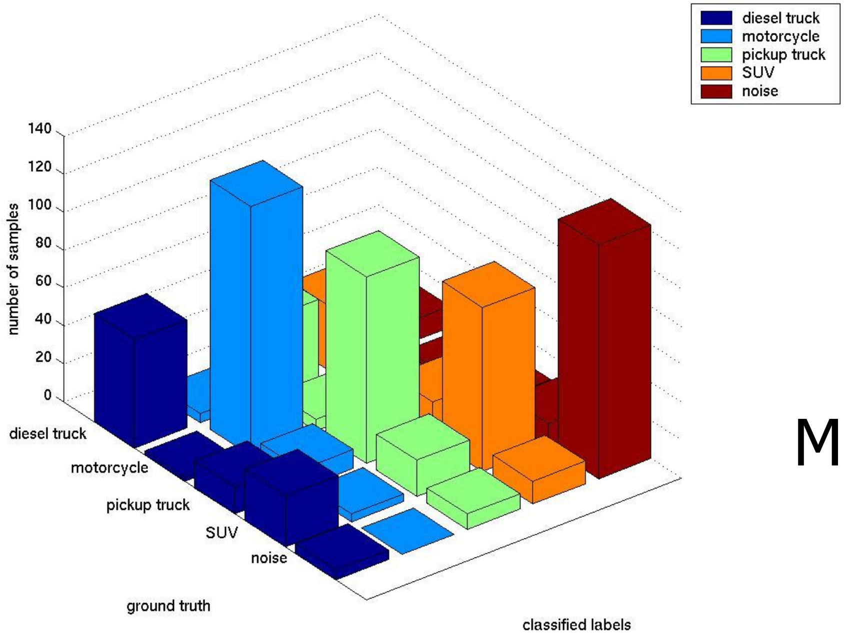
Multi-sensor fusion (96.44%)



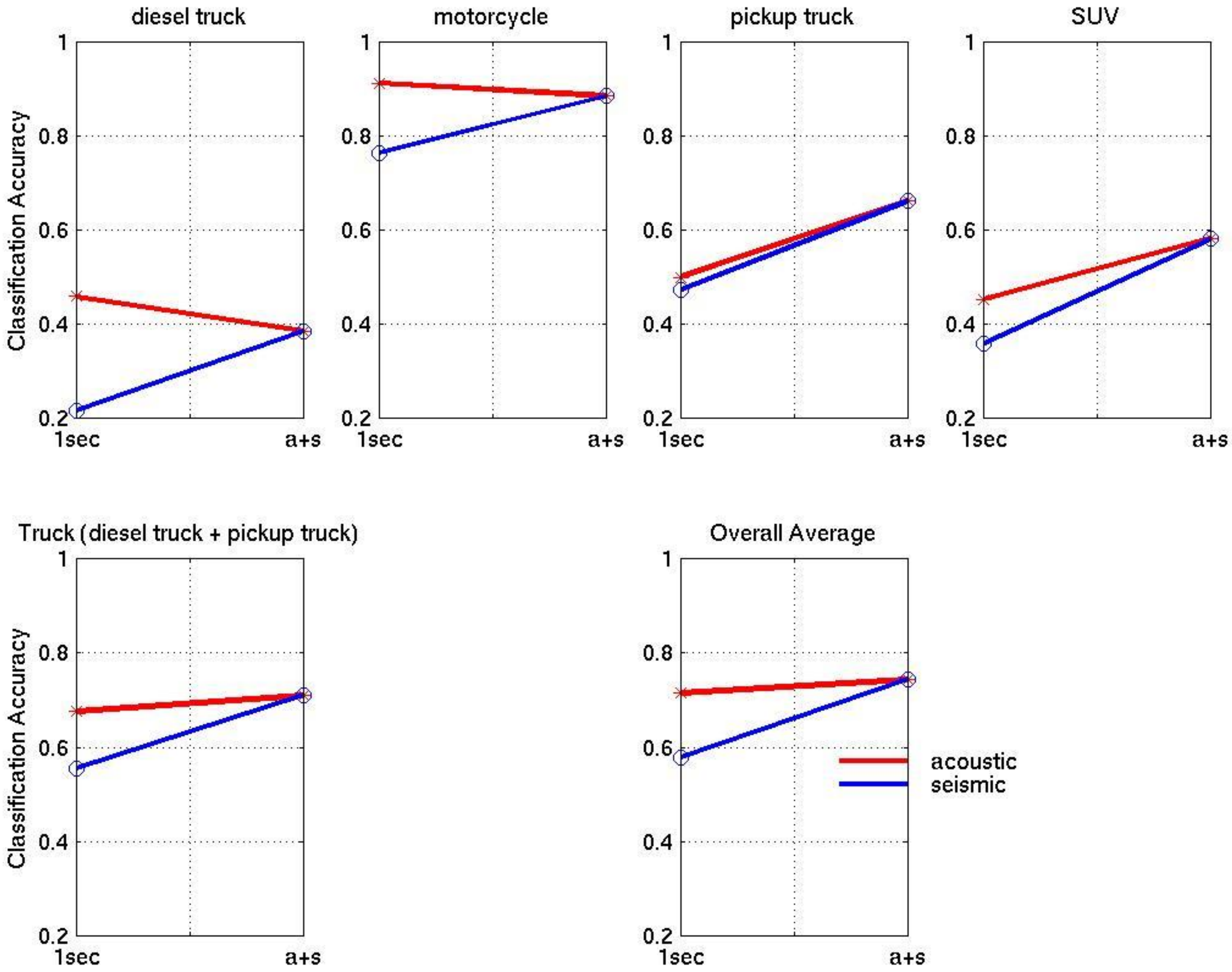
Acoustic



Seismic



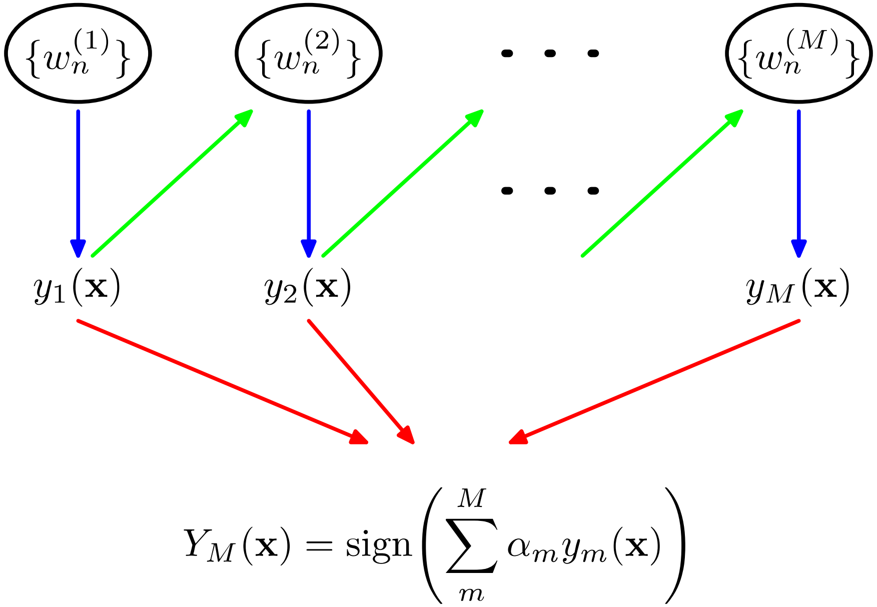
Multi-modal


diesel truck
motorcycle
pickup truck
SUV
Truck (diesel truck + pickup truck)
Overall Average
Classification Accuracy
1
0.8
0.6
0.4
0.2
1sec
a+s
acoustic
seismic

PART III: BOOSTING

Boosting

- Base classifiers are trained in sequence!
- Base classifiers as weak learners
- Weighted majority voting to combine classifiers



AdaBoost

- Step 1: Initialize the data weighting coefficients $\{w_n\}$ by setting $w_n^{(1)} = 1/N$, where N is the # of samples
- Step 2: for each classifier $y_m(\mathbf{x})$
  - (a) Fit a classifier $y_m(\mathbf{x})$ to the training data by minimizing the weighted error function

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)$$

  - (b) Evaluate the quantities

$$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}} \qquad \alpha_m = \ln\left\{\frac{1-\epsilon_m}{\epsilon_m}\right\}$$

  - (c) Update the data weighting coefficients

$$w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(y_m(\mathbf{x}_n) \neq t_n)\}$$

- Step 3: Make predictions using the final model

$$Y_M(\mathbf{x}) = \operatorname{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x})\right)$$

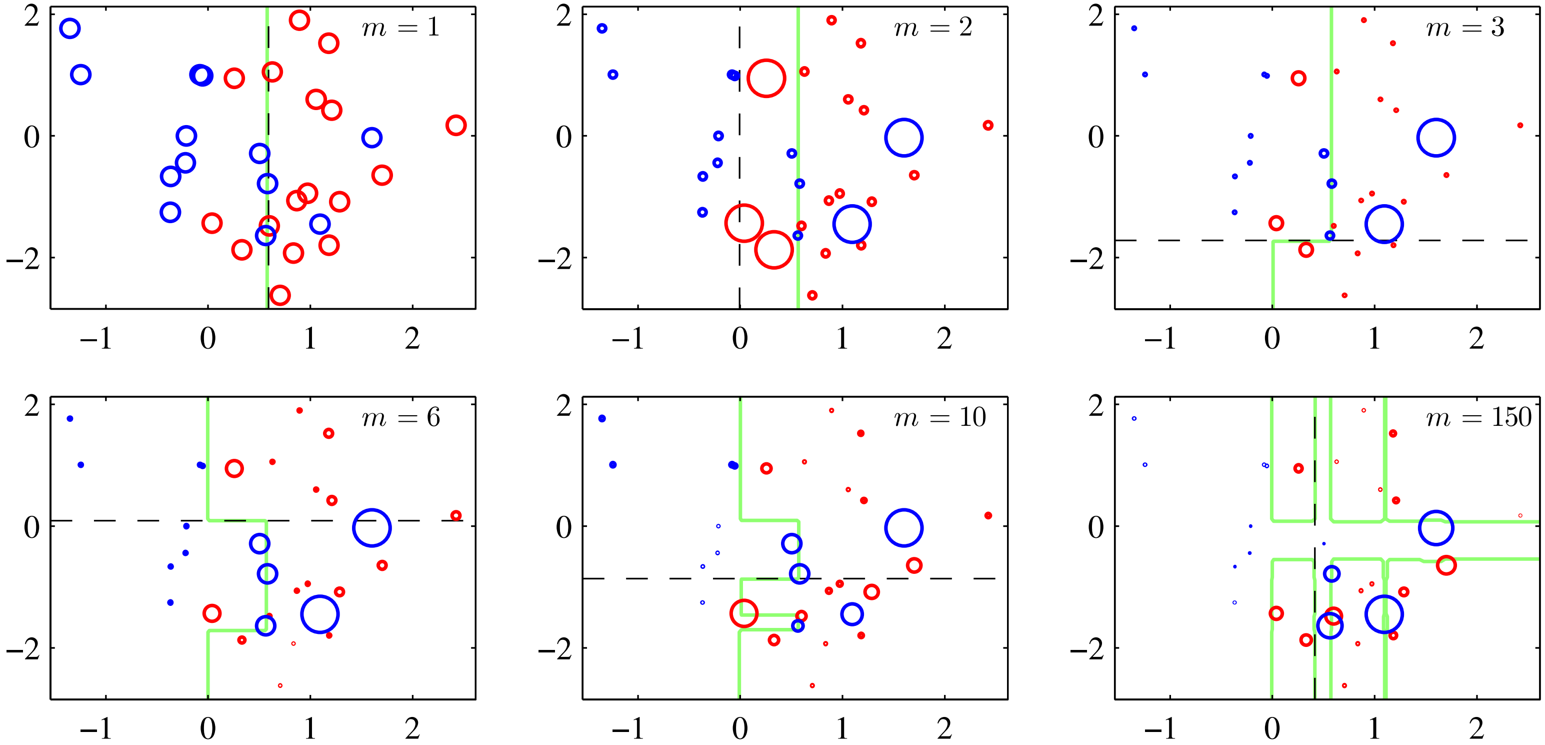
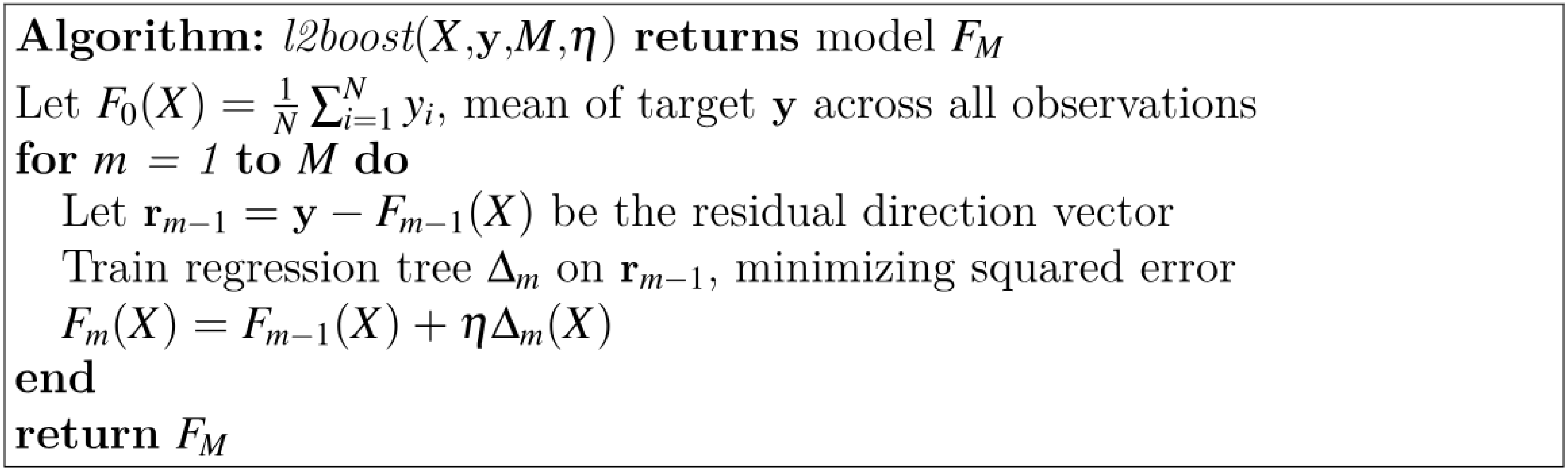


Figure 14.2 Illustration of boosting in which the base learners consist of simple thresholds applied to one or other of the axes. Each figure shows the number m of base learners trained so far, along with the decision boundary of the most recent base learner (dashed black line) and the combined decision boundary of the ensemble (solid green line). Each data point is depicted by a circle whose radius indicates the weight assigned to that data point when training the most recently added base learner. Thus, for instance, we see that points that are misclassified by the $m = 1$ base learner are given greater weight when training the $m = 2$ base learner.

Gradient Boosting



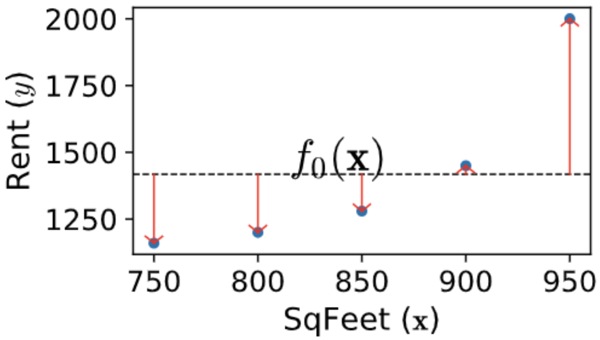
```
**Algorithm:** $\mathit{l2boost}(X,\mathbf{y},M,\eta)$ **returns** model $F_M$
Let $F_0(X) = \frac{1}{N}\sum_{i=1}^{N} y_i$, mean of target $\mathbf{y}$ across all observations
**for** $m$ = $\mathit{1}$ **to** $M$ **do**
    Let $\mathbf{r}_{m-1} = \mathbf{y} - F_{m-1}(X)$ be the residual direction vector
    Train regression tree $\Delta_m$ on $\mathbf{r}_{m-1}$, minimizing squared error
    $F_m(X) = F_{m-1}(X) + \eta\Delta_m(X)$
**end**
**return** $F_M$
```

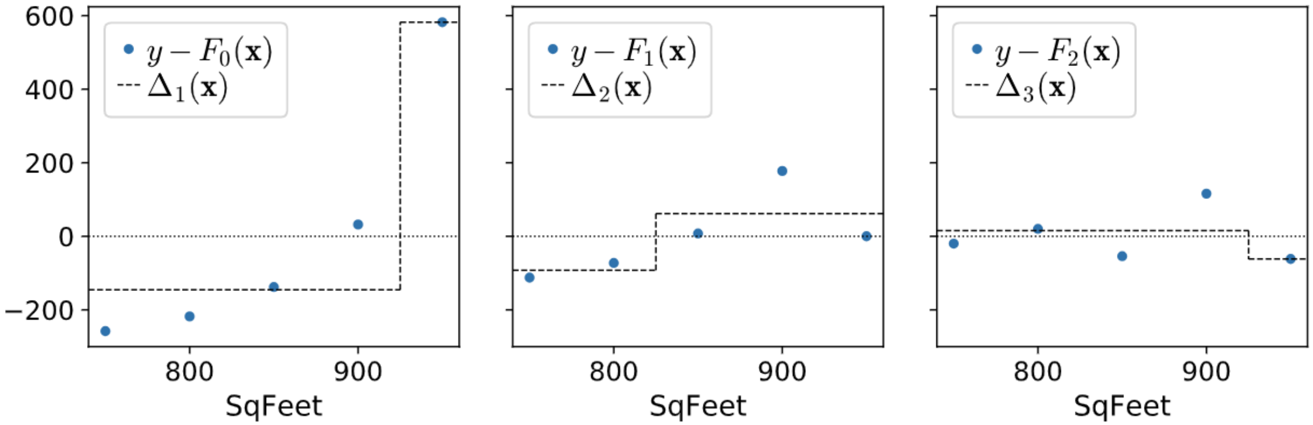
A Toy Example

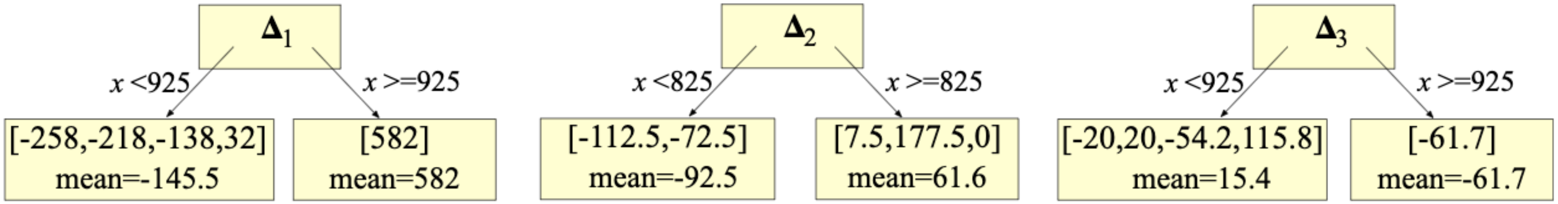
sqfeet	rent
750	1160
800	1200
850	1280
900	1450
950	2000

sqfeet	rent	F_0	$\mathbf{y} - F_0$
750	1160	1418	-258
800	1200	1418	-218
850	1280	1418	-138
900	1450	1418	32
950	2000	1418	582

Δ_1	F_1	y-F_1	Δ_2	F_2	y - F_2	Δ_3	F_3
-145.5	1272.5	-112.5	-92.5	1180	-20	15.4	1195.4
-145.5	1272.5	-72.5	-92.5	1180	20	15.4	1195.4
-145.5	1272.5	7.5	61.7	1334.2	-54.2	15.4	1349.6
-145.5	1272.5	177.5	61.7	1334.2	115.8	15.4	1349.6
582	2000	0	61.7	2061.7	-61.7	-61.7	2000







Δ_1

x <925

x >=925

[-258,-218,-138,32]
mean=-145.5

[582]
mean=582

Δ_2

x <825

x >=825

[-112.5,-72.5]
mean=-92.5

[7.5,177.5,0]
mean=61.6

Δ_3

x <925

x >=925

[-20,20,-54.2,115.8]
mean=15.4

[-61.7]
mean=-61.7

Reference

- For details regarding majority voting and Naïve Bayes, see http://www.cs.rit.edu/~nan2563/combining_classifiers_notes.pdf
- (!!)Terence Pass and Jeremy Howard, “How to explain gradient boosting”, https://explained.ai/gradient-boosting/index.html