BITS F464: Machine Learning

27 Boosting Bagging

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

March 28, 2021 ONLINE (Campus @ BITS-Pilani Jan-May 2021)

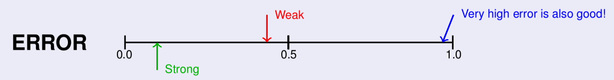
http://ktiwari.in/ml

Classification

- Binary classification approaches
  - Linear classifiers
  - Quadratic classifiers
  - Bayesian classification
  - Support vector machines
  - Decision trees
  - Neural networks
  - k-nearest neighbor

Which one to use?

Weak and strong classifier?



Boosting

- Can an ensemble of weak learners get a strong one

$$H(x) = sign(h^1(x) + h^2(x) + h^3(x) + ...)$$

- What is error rate?

$$\epsilon = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) \neq y^{(i)}) = \sum_{wrong}\frac{1}{m}$$

Three weak learners, having error on different data points is prefect

- But, most likely it is not going to be the case
- We can do the following
  1. Get weak learner h^1 (that is best)
  2. Exaggerate the data where h^1 errs and get weak learner h^2
  3. Exaggerate again the data for $h^1 \neq h^2$ and get weak learner h^3
- Will this work?

Boosting

- Boosting uses ensemble of learners (weak ones)
- AdaBoost[1] is one of the widely used boosting algorithm
- Given with m labeled training examples $(x^{(1)}, y^{(1)}), ..., (x^{(m)}, y^{(m)})$ where the $x^{(i)} \in \mathcal{X}$, and the labels $y^{(i)} \in \{-1, +1\}$
- Round t computes a distribution D_t over training examples. Weak learning algorithm is applied to find a suitable (low weighted error ϵ_t relative to D_t) weak hypothesis $h_t : \mathcal{X} \rightarrow \{-1, +1\}$
- Final or combined hypothesis H computes the sign of a weighted combination of weak hypotheses

$$H(x) = sign(\sum_{t=1}^{T}\alpha_t h_t(x))$$

[1] A decision-theoretic generalization of on-line learning and an application to boosting, Freund, Yoav and Schapire, Robert E, European conference on computational learning theory, pp 23–37 Springer(1995)

Boosting

- Let each data point have a weight w_i associated with them. Initially $w_i = \frac{1}{m}$ so that error rate becomes

$$\epsilon = \sum_{wrong}\frac{1}{m} = \sum_{i \in wrong} w_i$$

- Sum of weights is 1

$$\sum w_i = 1$$

- Weights are modified in every round

AdaBoost Algorithm

Algorithm 1: AdaBoost

```
Initialize w_1, ..., w_m to 1/m
for round t = 1 to T do
    Choose hypothesis h^t minimizing error ε^t on current distribution
    Compute α_t = 1/2 ln((1-ε^t)/ε^t)
    Update all w_1, ..., w_m values
return h and α
```

- Update of weights w_i at time t is done using

$$w_i^{t+1} = \frac{w_i^t}{z_t}e^{-(\alpha_t h^t(x^{(i)})y^{(i)})}$$

for all data point *i.e.* value of i varies from 1 to m

- Here z_t is a normalization factor

Normalization factor

- Here z_t is a normalization factor so

$$
\begin{aligned}
z_t &= \sum_{i=1}^{m} w_i^t e^{-(\alpha_t h^t(x^{(i)}) y^{(i)})} \\
&= \sum_{i \in right} w_i^t e^{-(\alpha_t h^t(x^{(i)}) y^{(i)})} + \sum_{i \in wrong} w_i^t e^{-(\alpha_t h^t(x^{(i)}) y^{(i)})} \\
&= \sum_{i \in right} w_i^t e^{-(\alpha_t)} + \sum_{i \in wrong} w_i^t e^{-(\alpha_t(-1))} \\
&= \sum_{i \in right} w_i^t \sqrt{\frac{\epsilon^t}{1-\epsilon^t}} + \sum_{i \in wrong} w_i^t \sqrt{\frac{1-\epsilon^t}{\epsilon^t}} \\
&= \sqrt{\frac{\epsilon^t}{1-\epsilon^t}} \sum_{i \in right} w_i^t + \sqrt{\frac{1-\epsilon^t}{\epsilon^t}} \sum_{i \in wrong} w_i^t \\
&= \sqrt{\frac{\epsilon^t}{1-\epsilon^t}} (1-\epsilon^t) + \sqrt{\frac{1-\epsilon^t}{\epsilon^t}} \epsilon^t \\
&= 2\sqrt{\epsilon^t(1-\epsilon^t)}
\end{aligned}
$$

Boosting

- Weight update for correctly classified data point

$$w_i^{t+1} = w_i^t \frac{1}{2(1-\epsilon^t)}$$

- Weight update is therefore for wrongly classified data point

$$w_i^{t+1} = w_i^t \frac{1}{2\epsilon^t}$$

Final hypothesis

$$H(x) = sign(\alpha_1 h^1(x) + \alpha_2 h^2(x) + \alpha_3 h^3(x) + \ldots)$$

$$\epsilon^t = \sum_{i \in wrong} w_i^t$$

AdaBoost at work

Consider the following data set

	(x	y)
s_1	(1,	-1)
s_2	(2,	-1)
s_3	(3,	+1)
s_4	(4,	+1)
s_5	(5,	+1)
s_6	(6,	+1)
s_7	(7,	+1)
s_8	(8,	-1)
s_9	(9,	-1)
s_{10}	(10,	-1)

- **Round-00:** initialize weights

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

- **Round-01:** Let sampling according to w produces $(3, 10, 8, 2, 7, 5, 1, 3, 8, 3)$ so following sub-set of data is considered

s_1	(1,	-1)
s_2	(2,	-1)
s_3	(3,	+1)
s_5	(5,	+1)
s_7	(7,	+1)
s_8	(8,	-1)
s_{10}	(10,	-1)

Consider various hypothesis

AdaBoost at work (Round-01)

			h_a	h_b	h_c	h_d	h_e	h_f	h_g	h_h
s_1	(1,	-1)	+1	-1	-1	-1	-1	-1	-1	-1
s_2	(2,	-1)	+1	+1	-1	-1	-1	-1	-1	-1
s_3	(3,	+1)	+1	+1	+1	-1	-1	-1	-1	-1
s_5	(5,	+1)	+1	+1	+1	+1	-1	-1	-1	-1
s_7	(7,	+1)	+1	+1	+1	+1	+1	-1	-1	-1
s_8	(8,	-1)	+1	+1	+1	+1	+1	+1	-1	-1
s_{10}	(10,	-1)	+1	+1	+1	+1	+1	+1	+1	-1

- Select h_c as $h1$
- What is decision threshold? 2.5
- Compute error on whole dataset

AdaBoost at work (Round-01)

Threshold is 2.5

	(x	y)	h_1
s_1	(1,	-1)	-1
s_2	(2,	-1)	-1
s_3	(3,	+1)	+1
s_4	(4,	+1)	+1
s_5	(5,	+1)	+1
s_6	(6,	+1)	+1
s_7	(7,	+1)	+1
s_8	(8,	-1)	+1
s_9	(9,	-1)	+1
s_{10}	(10,	-1)	+1

- Error rate $\epsilon = \sum_{i \in wrong} w_i$ $= w_8 + w_9 + w_{10} = 0.1 + 0.1 + 0.1 = 0.3$
- $\alpha = \frac{1}{2} \ln\left(\frac{1-\epsilon}{\epsilon}\right) = \frac{1}{2} \ln\left(\frac{1-0.3}{0.3}\right) = 0.4236$
- Weights are modified according to

$$w_i = w_i \times \begin{cases} \frac{1}{2(1-\epsilon)} = 0.7142 & \text{correct} \\ \frac{1}{2\epsilon} = 1.6666 & \text{wrong} \end{cases}$$

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.17	0.17	0.17

AdaBoost at work (Round-02)

Consider the data set

	(x	y)
s_1	(1,	-1)
s_2	(2,	-1)
s_3	(3,	+1)
s_4	(4,	+1)
s_5	(5,	+1)
s_6	(6,	+1)
s_7	(7,	+1)
s_8	(8,	-1)
s_9	(9,	-1)
s_{10}	(10,	-1)

- **Round-02:** Let sampling according to new w produces $(8, 10, 7, 10, 3, 1, 10, 8, 4, 9)$ so following sub-set of data is considered

s_1	(1,	-1)
s_3	(3,	+1)
s_4	(4,	+1)
s_7	(7,	+1)
s_8	(8,	-1)
s_9	(9,	-1)
s_{10}	(10,	-1)

Consider various hypothesis

AdaBoost at work (Round-02)

	x	y	h_a	h_b	h_c	h_d	h_e	h_f	h_g	h_h
s_1	(1,	-1)	+1	-1	-1	-1	-1	-1	-1	-1
s_3	(3,	+1)	+1	+1	-1	-1	-1	-1	-1	-1
s_4	(4,	+1)	+1	+1	+1	-1	-1	-1	-1	-1
s_7	(7,	+1)	+1	+1	+1	+1	-1	-1	-1	-1
s_8	(8,	-1)	+1	+1	+1	+1	+1	-1	-1	-1
s_9	(9,	-1)	+1	+1	+1	+1	+1	+1	-1	-1
s_{10}	(10,	-1)	+1	+1	+1	+1	+1	+1	+1	-1

- Select h_h as $h2$
- What is decision threshold? 10.5
- Compute error on whole dataset

AdaBoost at work (Round-02)

Threshold is 10.5

	x	y	h_2
s_1	(1,	-1)	-1
s_2	(2,	-1)	-1
s_3	(3,	+1)	-1
s_4	(4,	+1)	-1
s_5	(5,	+1)	-1
s_6	(6,	+1)	-1
s_7	(7,	+1)	-1
s_8	(8,	-1)	-1
s_9	(9,	-1)	-1
s_{10}	(10,	-1)	-1

- Error rate $\epsilon = \sum_{i \in wrong} w_i$
  $= w_3 + w_4 + w_5 + w_6 + w_7 = 5 \times 0.07 = 0.35$
- $\alpha = \frac{1}{2} \ln\left(\frac{1-\epsilon}{\epsilon}\right) = \frac{1}{2} \ln\left(\frac{1-0.35}{0.35}\right) = 0.3095$
- Weights are modified according to

$$w_i = w_i \times \begin{cases} \frac{1}{2(1-\epsilon)} = 0.7692 & \text{correct} \\ \frac{1}{2\epsilon} = 1.4285 & \text{wrong} \end{cases}$$

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.17	0.17	0.17
0.037	0.037	0.177	0.177	0.177	0.177	0.177	0.091	0.091	0.091

AdaBoost at work (Round-03)

Consider the data set

	x	y
s_1	(1,	-1)
s_2	(2,	-1)
s_3	(3,	+1)
s_4	(4,	+1)
s_5	(5,	+1)
s_6	(6,	+1)
s_7	(7,	+1)
s_8	(8,	-1)
s_9	(9,	-1)
s_{10}	(10,	-1)

- **Round-03:** Let sampling according to new w produces $(8, 7, 4, 8, 6, 9, 5, 8, 3, 4)$ so following sub-set of data is considered

	x	y
s_3	(3,	+1)
s_4	(4,	+1)
s_5	(5,	+1)
s_6	(6,	+1)
s_7	(7,	+1)
s_8	(8,	-1)
s_9	(9,	-1)

AdaBoost at work (Round-03)

	x	y	h_a	h_b	h_c	h_d	h_e	h_f	h_g	h_h
s_3	(3,	+1)	+1	-1	-1	-1	-1	-1	-1	-1
s_4	(4,	+1)	+1	+1	-1	-1	-1	-1	-1	-1
s_5	(5,	+1)	+1	+1	+1	-1	-1	-1	-1	-1
s_6	(6,	+1)	+1	+1	+1	+1	-1	-1	-1	-1
s_7	(7,	+1)	+1	+1	+1	+1	+1	-1	-1	-1
s_8	(8,	-1)	+1	+1	+1	+1	+1	+1	-1	-1
s_9	(9,	-1)	+1	+1	+1	+1	+1	+1	+1	-1

- Select h_a as $h3$
- What is decision threshold? 0.5
- Compute error on whole dataset

AdaBoost at work (Round-03)

Threshold is 0.5

	x	y	h_3
s_1	(1,	-1)	+1
s_2	(2,	-1)	+1
s_3	(3,	+1)	+1
s_4	(4,	+1)	+1
s_5	(5,	+1)	+1
s_6	(6,	+1)	+1
s_7	(7,	+1)	+1
s_8	(8,	-1)	+1
s_9	(9,	-1)	+1
s_{10}	(10,	-1)	+1

- Error rate $\epsilon = \sum_{i \in wrong} w_i$
  $= 2 \times 0.037 + 3 \times 0.091 = 0.34$
- $\alpha = \frac{1}{2} \ln\left(\frac{1-\epsilon}{\epsilon}\right) = \frac{1}{2} \ln\left(\frac{1-0.34}{0.34}\right) = 0.3316$
- We may continue for next round like that
- Let use see our accuracy now

AdaBoost at work (Round-03)

$$(\alpha_1, \alpha_2, \alpha_3) = (0.4236, 0.3095, 0.3316)$$

	x	y	h_1	h_2	h_3	$sign(\sum_i(\alpha_i \times h_i))$
s_1	(1,	-1)	-1	-1	+1	-1
s_2	(2,	-1)	-1	-1	+1	-1
s_3	(3,	+1)	+1	-1	+1	+1
s_4	(4,	+1)	+1	-1	+1	+1
s_5	(5,	+1)	+1	-1	+1	+1
s_6	(6,	+1)	+1	-1	+1	+1
s_7	(7,	+1)	+1	-1	+1	+1
s_8	(8,	-1)	+1	-1	+1	+1
s_9	(9,	-1)	+1	-1	+1	+1
s_{10}	(10,	-1)	+1	-1	+1	+1

AdaBoost Summary

Algorithm 2: AdaBoost

1. Initialize $w_1, ..., w_m$ to $\frac{1}{m}$
2. **for** *round* $t = 1$ *to* T **do**
3. Choose hypothesis h^t minimizing error ϵ^t on current distribution
4. Compute $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon^t}{\epsilon^t}\right)$
5. Update all $w_1, ..., w_m$ values
6. **return** h and α

$$\epsilon^t = \sum_{i \in wrong} w_i^t$$

$$w_i^{t+1} = \frac{w_i^t}{z_t} e^{-(\alpha_t h^t(x^{(i)}) y^{(i)})}$$

$$z_t = 2\sqrt{\epsilon^t(1-\epsilon^t)}$$

$$w_i^{t+1} = \begin{cases} w_i^t \frac{1}{2(1-\epsilon^t)} & \text{correctly classified data point} \\ w_i^t \frac{1}{2\epsilon^t} & \text{otherwise} \end{cases}$$

Final hypothesis

$$H(x) = sign(\alpha_1 h^1(x) + \alpha_2 h^2(x) + \alpha_3 h^3(x) + ...)$$

Bagging

- A technique that repeatedly samples (with replacement) m training data points according the uniform distribution
- Some data points may be repeated or omitted. Sample set is expected to have 63% of items [2]

Algorithm 3: Bagging

1. Let k be the number of bootstrap samples
2. **for** $i = 1$ *to* k **do**
3. Create D_i; a bootstrap sample of size m
4. Train classifier C_i on D_i
5. **return** C_i's

$$C(x) = \underset{y}{\text{argmax}} \sum_i \delta(C_i(x) = y) \qquad \delta(true) = 1; \textit{othewise } 0$$

[2] An item has the probability $1 - (1 - \frac{1}{m})^m$ of being selected that converges to $1 - \frac{1}{e}$ as m increases

Bagging Example

Consider the following data set

	(x	y)
s_1	(0.1,	+1)
s_2	(0.2,	+1)
s_3	(0.3,	+1)
s_4	(0.4,	-1)
s_5	(0.5,	-1)
s_6	(0.6,	-1)
s_7	(0.7,	-1)
s_8	(0.8,	+1)
s_9	(0.9,	+1)
s_{10}	(1.0,	+1)

Let sampling gets data points as s_1, s_2, s_3, s_4, s_5, s_6, s_9, s_{10} Then

	(x y)	0	.15	.25	.35	.45	.55	.65	.75	1.05
s_1	(0.1,+1)	+1	-1	-1	-1	-1	-1	-1	-1	-1
s_2	(0.2,+1)	+1	+1	-1	-1	-1	-1	-1	-1	-1
s_3	(0.3,+1)	+1	+1	+1	-1	-1	-1	-1	-1	-1
s_4	(0.4,-1)	+1	+1	+1	+1	-1	-1	-1	-1	-1
s_5	(0.5,-1)	+1	+1	+1	+1	+1	-1	-1	-1	-1
s_6	(0.6,-1)	+1	+1	+1	+1	+1	+1	-1	-1	-1
s_9	(0.9,+1)	+1	+1	+1	+1	+1	+1	+1	-1	-1
s_{10}	(1.0,+1)	+1	+1	+1	+1	+1	+1	+1	+1	-1

Best classifier can be obtained by taking threshold at 0.35 and reversing the sign.

$$Classification = \begin{cases} +1 & \text{if } x \leq 0.35 \\ -1 & \text{otherwise} \end{cases}$$

Bagging Example

Similarly

	(x	y)
s_1	(0.1,	+1)
s_2	(0.2,	+1)
s_3	(0.3,	+1)
s_4	(0.4,	-1)
s_5	(0.5,	-1)
s_6	(0.6,	-1)
s_7	(0.7,	-1)
s_8	(0.8,	+1)
s_9	(0.9,	+1)
s_{10}	(1.0,	+1)

Bagging Round 1:

x	0.1	0.2	0.3	0.4	0.5	0.6	0.9
y	1	1	1	-1	-1	-1	1

x <= 0.35 ==> y = 1
x > 0.35 ==> y = -1

Bagging Round 2:

x	0.1	0.2	0.3	0.4	0.5	0.8	0.9	1
y	1	1	1	-1	-1	1	1	1

x <= 0.65 ==> y = 1
x > 0.65 ==> y = 1

Bagging Round 3:

x	0.1	0.2	0.3	0.4	0.5	0.7	0.8	0.9
y	1	1	1	-1	-1	-1	1	1

x <= 0.35 ==> y = 1
x > 0.35 ==> y = -1

Bagging Round 4:

x	0.1	0.2	0.4	0.5	0.7	0.8	0.9
y	1	1	-1	-1	-1	1	1

x <= 0.3 ==> y = 1
x > 0.3 ==> y = -1

Bagging Round 5:

x	0.1	0.2	0.5	0.6	1
y	1	1	-1	-1	1

x <= 0.35 ==> y = 1
x > 0.35 ==> y = -1

Bagging Round 6:

x	0.2	0.4	0.5	0.6	0.7	0.8	0.9	1
y	1	-1	-1	-1	-1	1	1	1

x <= 0.75 ==> y = -1
x > 0.75 ==> y = 1

Bagging Round 7:

x	0.1	0.4	0.6	0.7	0.8	0.9	1
y	1	-1	-1	-1	1	1	1

x <= 0.75 ==> y = -1
x > 0.75 ==> y = 1

Bagging Round 8:

x	0.1	0.2	0.5	0.7	0.8	0.9	1
y	1	1	-1	-1	1	1	1

x <= 0.75 ==> y = -1
x > 0.75 ==> y = 1

Bagging Round 9:

x	0.1	0.3	0.4	0.6	0.7	0.8	1
y	1	1	-1	-1	-1	1	1

x <= 0.75 ==> y = -1
x > 0.75 ==> y = 1

Bagging Round 10:

x	0.1	0.3	0.8	0.9
y	1	1	1	1

x <= 0.05 ==> y = -1
x > 0.05 ==> y = 1

Bagging Example

x <= 0.35 ==> y = 1
x > 0.35 ==> y = -1

x <= 0.65 ==> y = 1
x > 0.65 ==> y = 1

x <= 0.35 ==> y = 1
x > 0.35 ==> y = -1

x <= 0.3 ==> y = 1
x > 0.3 ==> y = -1

x <= 0.35 ==> y = 1
x > 0.35 ==> y = -1

x <= 0.75 ==> y = -1
x > 0.75 ==> y = 1

x <= 0.75 ==> y = -1
x > 0.75 ==> y = 1

x <= 0.75 ==> y = -1
x > 0.75 ==> y = 1

x <= 0.75 ==> y = -1
x > 0.75 ==> y = 1

x <= 0.05 ==> y = -1
x > 0.05 ==> y = 1

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Round	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	1	1	1	-1	-1	-1	-1	-1	-1	-1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	-1	-1	-1	-1	-1	-1	-1
4	1	1	1	-1	-1	-1	-1	-1	-1	-1
5	1	1	1	-1	-1	-1	-1	-1	-1	-1
6	-1	-1	-1	-1	-1	-1	-1	1	1	1
7	-1	-1	-1	-1	-1	-1	-1	1	1	1
8	-1	-1	-1	-1	-1	-1	-1	1	1	1
9	-1	-1	-1	-1	-1	-1	-1	1	1	1
10	1	1	1	1	1	1	1	1	1	1
Sum	2	2	2	-6	-6	-6	-6	2	2	2
Sign	1	1	1	-1	-1	-1	-1	1	1	1
True Class	1	1	1	-1	-1	-1	-1	1	1	1

Thank You!

Thank you very much for your attention!

Queries ?

(Reference[3])

[3] [1] A desicion-theoretic generalization of on-line learning and an application to boosting, Freund, Yoav and Schapire, Robert E, European conference on computational learning theory, pp 23–37 Springer(1995) [2] Learning: Boosting https://www.youtube.com/watch?v=UHBmv7qCey4 Patrick Winston