BITS F464: Machine Learning

03 Performance Evaluation

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

Jan 22, 2021 ONLINE (Campus @ BITS-Pilani Jan-May 2021)

http://ktiwari.in/ml

Recap: ML Building Blocks

- Input: x
- Output: y
- Training data: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(m)}, y^{(m)})$
- $x^{(i)}$ could be a multivariate say $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, ..., x_n^{(i)})$
- Concept, target function: true function

$$f : x \rightarrow y$$

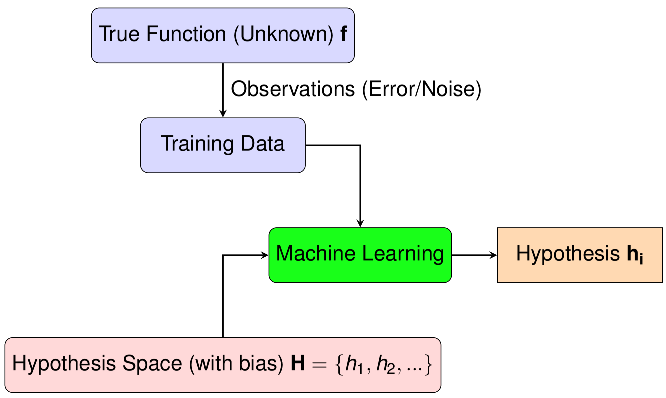
- Hypothesis:

$$h : x \rightarrow y$$

- Accuracy: agreement b/w f and h

Issue is
The true function is NOT known.

Recap: The Flow of ML



Recap: A Toy model

- **The Problem:** credit approval.
- Input: $x = (x_1, x_2, ..., x_n)$
- Let x_1=accountBal, x_2=Salary, x_3=age ...
- What weights we should give w_1=0.6, w_2=0.3, w_3=-0.1 ...
- The Model

$$\sum_{i=1}^{n} w_i \times x_i = \begin{cases} > \textit{Threshold} & \text{Then APPROVE} \\ \textit{otherwise} & \text{DENY/REJECT} \end{cases}$$

- Simplified:

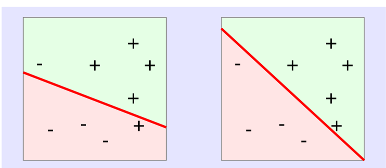
$$h(x) = sign(\sum_{i=1}^{n} w_i \times x_i - Threshold)$$

- Add an extra term x_0 (that is always 1), then

$$h(x) = sign(\sum_{i=0}^{n} w_i \times x_i) = sign(w^T x)$$

Recap: A Toy model (Contd..)

- Can you recognize $h(x) = sign(\sum_{i=0}^{n} w_i \times x_i)$
- It is a linear equation (in two dimension) or hyper plane
- Sign could be positive or negative, so two classes are **+1** and **–1**



- Vector $w = (w_0, w_2, ..., w_n)$ would be normal to the plane of linear decision boundary. (why? because dot product is $\cos\theta$)
- What could change this plane? w_i's

Learning: Use misclassified examples to update $w_i = w_i + \alpha y_i x_i$

Recap: A Toy model (What $yw^T x$ tells)

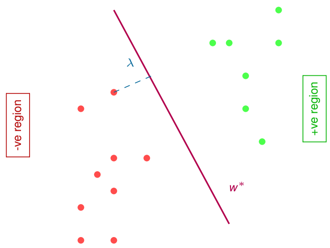
- Classification of a point x can be obtained by $w^T x$. If $w^T x$ if positive then x is positive, otherwise negative.
- ML assumes that data point is never on hyperplane so $w^T x \neq 0$
- There could be two cases
  1. **When classification of the model is correct:**
     For $y = +1$ we have $w^T x > 0$
     For $y = -1$ we have $w^T x < 0$
     In both the cases $yw^T x > 0$
  2. **When classification of the model is wrong:**
     For $y = +1$ we have $w^T x < 0$
     For $y = -1$ we have $w^T x > 0$
     In both the cases $yw^T x < 0$

So we have a simple test

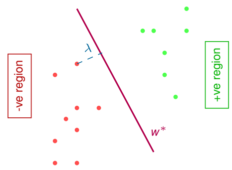
$$yw^T x \begin{cases} > 0 & \text{Classification is correct} \\ < 0 & \text{Classification is wrong} \end{cases}$$

Conversion Proof

IF *data is linearly separable*; then the learning rule $w_i = w_i + \alpha y_i x_i$ converges in finitely many steps



Conversion Proof



- Let w^* be separating hyperplane
- Make $||w^*|| = 1$, $\forall_i ||x_i|| \le 1$
- Let nearest data point be λ distance apart from w^*
- Consider $y{w^*}^T x$ it is > 0 for all $x \in D$ with minimum value λ

Update increases $w^T w^*$ least by $\alpha\lambda$

$$(w + \alpha yx)^T w^* = w^T w^* + \alpha y {w^*}^T x \ge w^T w^* + \alpha\lambda$$

$w.w^T$ increases less then α^2

$(w + \alpha yx).(w + \alpha yx)^T = w.w^T + 2\alpha y.w^T x + \alpha^2 y^2 x.x^T$
x is used to update as it is mis classified so $y.w^T x < 0$. Also $x.x^T < 1$ due to scaling and $y^2 = 1$. So, $(w + \alpha yx) \times (w + \alpha yx)^T \le w.w^T + \alpha^2$

Conversion Proof: After k updates

$$(w + \alpha yx)^T w^* \ge w^T w^* + \alpha\lambda \quad (1)$$

$$(w + \alpha yx).(w + \alpha yx)^T \le w.w^T + \alpha^2 \quad (2)$$

After k updates (when it converses)

$$\begin{aligned} k.\alpha\lambda &\le w^T.w^* && \text{using eq-1 for } k \text{ number of times} \\ &= ||w^T.w^*|| \le ||w^T||.||w^*|| && \text{Cauchy Schwarz Inequality} \\ &= ||w|| && \text{as } ||w^*|| = 1 \\ &= \sqrt{W^T W} < \sqrt{k\alpha^2} && \text{using eq-2 for } k \text{ number of times} \\ k.\alpha\lambda &\le \sqrt{k\alpha^2} && (3) \end{aligned}$$

$k \le 1/\lambda^2$ Number of update steps is bounded by $1/\lambda^2$

Loss Function

Performance is the closeness of hypothesis with target function

- For example
  - Classification

$$loss(y, h(x)) = \begin{cases} 1 & \text{if } h(x) \ne y \\ 0 & \text{otherwise} \end{cases}$$

  - Regression

$$loss(y, h(x)) = \begin{cases} (h(x) - y)^2 & \text{if } h(x) \ne y \\ 0 & \text{otherwise} \end{cases}$$

Issue is that, we can only do Empirical Risk Minimization.

Since only training data is available, on can use only this to be good (minimize risk on empirical data)

Statistics

There were 100 images in a box. 30 of them were containing lion. I asked Bob to separate all the pics of lion. He showed me 60 but, lion was not in 40 of them.

- True positives (TP): 20
- True negatives (TN): 30
- T1-Error: False positives (FP): 40
- T2-Error: False negatives (FN): 10

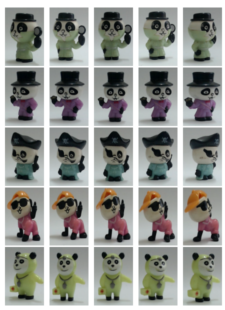
Confusion Matrix

Ground Truth \ Experiment	T	F
T	20	10
F	40	30

Accuracy: ((20+30)/100)*100%,
Precision: (20/60)*100%,
Recall (true positive rate or Sensitivity): (20/(20+10))*100%,
Specificity (true negative rate): (30/(40+30))*100%,
F Score: (Precision+Recall)/2,
F1 Measure: Harmonic mean of Precision and Recall

Matching Scores

Consider a system providing matching score between two images.

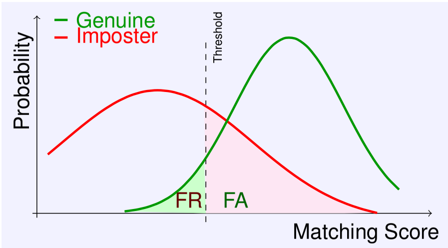


- Scores could be similarity or dissimilarity
- Matching could be Genuine or Imposter
- Input two images
- Output a score. normalized in [0,1]

ID1	P1	ID2	P2	G/I	Score
1	1	1	2	1	0.82
1	1	1	3	1	0.88
1	2	2	1	0	0.48
3	2	2	4	0	0.32
3	2	2	3	0	0.19
1	2	1	1	1	0.78
4	2	5	3	0	0.26
3	2	5	3	0	0.19
5	2	5	1	1	0.78
3	4	4	4	0	0.32
3	5	1	3	0	0.19
5	5	5	1	1	0.88
...	...	...	...	...	...

Matching Scores and Performance

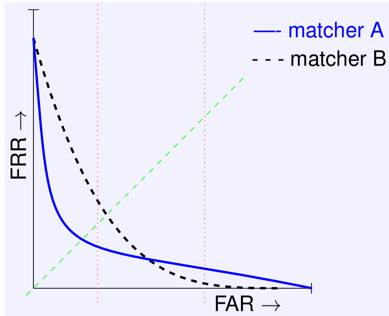
Matching score could either be similarity or dissimilarity.



Two type of Errors
Type-I: False Acceptance Rate (FAR), chance of accepting an intruder
Type-II: False Rejection Rate (FRR) chance of rejecting a genuine [1]

[1] When FAR increases, FRR decreases. Threshold is used to take decision on accept/reject.

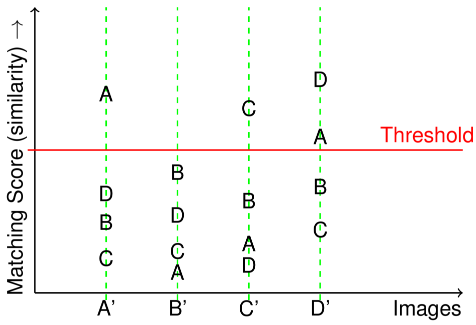
Receiver Operating Curve (ROC)



Equal error rate (EER) is a point where FAR and FRR are equal

Area under ROC represents error.

Error Happens

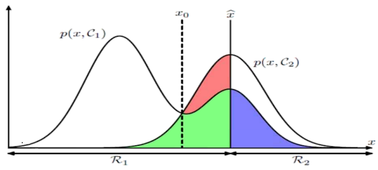


CRR is 100% but EER is ∼12%.

Minimize Misclassification

Goal is to minimize misclassification rate (risk)

$$p(mistake) = p(x \in R_1, C_2) + p(x \in R_2, C_1)$$

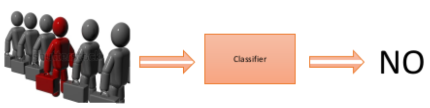


$$p(mistake) = \int_{R_1} p(x, C_2)\, dx + \int_{R_2} p(x, C_1)\, dx$$

Failure should be a part of model

Problem: *classifier for terrorists trying to board a flight*

- I can give you a 99.99% accurate model
- Claim that it is so simple that you can compute it in your head



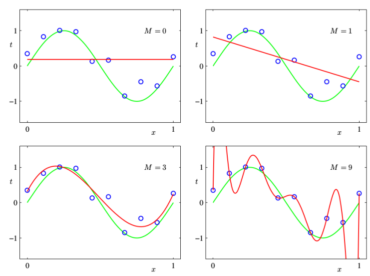
Fact: 800 million average passengers on US flights per year, 19 (confirmed) terrorists who boarded US flights from 2000–2017.

$$accuracy = 99.9999999\%$$

You cannot neglect which side the error is

Polynomial Curve Fitting (Towards Overfitting)

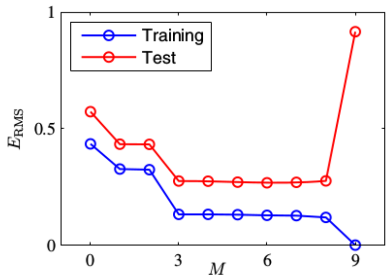
When is the curve $h_w(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_3 x^M$ better?



Learning attempts to minimize error $E(w) = \frac{1}{2}\sum_{n=1}^{N}(h_w(x_n) - y_n)^2$

Training and Test Error

Data is split in 1) Training 2) Testing 3) Validation [2]



Training error decreases with more complex model.
What happened at the end? (overfitting?)

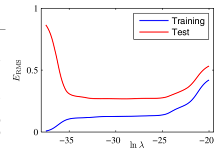
[2] Generally 70, 20 and 10 %

Regularization

Coefficient increases as the order of polynomial increases [3]

	$M = 0$	$M = 1$	$M = 6$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

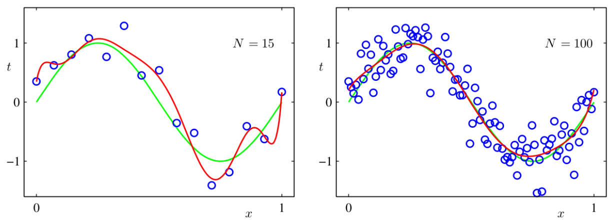


$$E(w) = \frac{1}{2}\sum_{n=1}^{N}(h_w(x_n) - y_n)^2 + \frac{\lambda}{2}||w||^2$$

where $||w|| = w^T w = w_0^2 + w_1^2 + w_2^2 + \ldots$

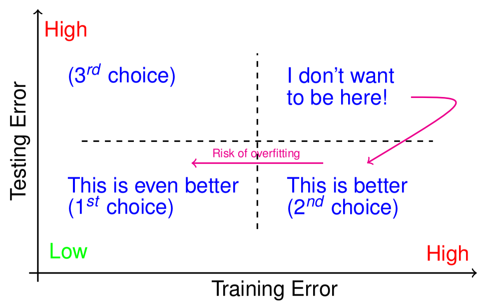
[3] Here λ is something like $1/e^{-\beta}$

Having more data also helps regularization



- Having more training examples reduce the overfitting problem
- We can train more complex models if we have more data

Which side do you want to be?



Low training error comes with a risk of overfitting.

Thank You!

Thank you very much for your attention!

Queries ?