Linear and non-linear classifiers

Tuesday, September 10, 2019 2:34 PM

- Machine Learning
  - ↳ Supervised Learning
    - ↳ Basic Models — Decision Trees
      - — linear and non-linear classifiers

Inspired by Numerical Methods :-

Regression Review.

Examples E $\langle X_1, X_2, X_3 \dots X_n, Y \rangle$ $X_i \in \mathbb{R}$, $Y \in \mathbb{R}$

We are trying to learn $\hat{Y} = (X_1, X_2, \dots X_n) : Y$

Linear regression.

Assume $\hat{Y} = w_0 + X_1 w_1 + X_2 w_2 + X_3 w_3 + \dots + X_n w_n$

where w_i are unknown.

find such w_i !!

- Introduce $X_0 = 1$ always. then:

$$\hat{Y} = \sum_{i=0}^{n} w_i X_i$$

Error Function :- to compare $\hat{Y}$ and Y

$$\text{Error}(E) = \sum_{e \in E} \left(Y(e) - \hat{Y}(e) \right)^2$$

Lets use sum of squares error

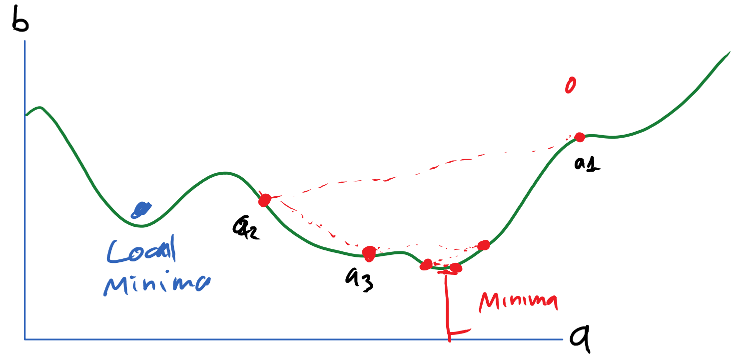
$$= \sum_{e \in E} \left(Y(e) - \sum_{i=0}^{n} w_i X_i(e) \right)^2$$

find w_i's that minimize Error (E)?

Technique: Gradient Descent

Iterative Method to find Minima of functions

cartoon version



function must be differentiable.

$b = f(a)$

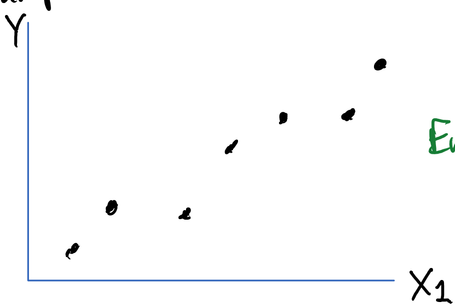
$$a_{n+1} \leftarrow a_n + \eta \cdot f'(a)$$

learning rate

derivative of f'

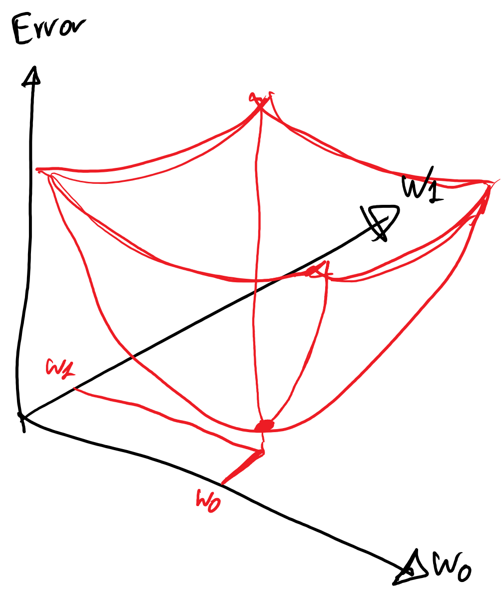
So: Apply Gradient Descent to our Error (E) function:

Examples:



$$\hat{Y} = w_0 + X_1 w_1$$

$$Error = \left(Y - \left(w_0 + X_1 w_1\right)\right)^2$$



- update rule:

$$w_i \leftarrow w_i - \eta * \frac{\partial}{\partial w_i} Error(E)$$

Learning Rate

If using Sum of square Errors

$$\frac{\partial}{\partial w_i} Error(E) = \sum_{e \in E} -2 \cdot \left(\underbrace{Y(e) - \hat{Y}(e)}\right) \cdot X_i(e)$$

$$\frac{\partial}{\partial w_i} Error(E) = \sum_{e \in E} -2 \cdot \underbrace{(Y(e) - \hat{Y}(e))}_{\delta(e)} \cdot X_i(e)$$

Compute for each w_i $\quad \frac{\partial}{\partial w_i} Error(E) = \sum_{e \in E} -2 \cdot \delta(e) \cdot X_i(e)$

update each w_i by $-\eta * \frac{\partial}{\partial w_i} Error(E)$

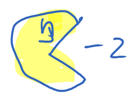
Repeat until stop criteria reached.

- $\delta(e)$'s become small
- changes to w_i become small.

- Variant: incremental Gradient Descent.
  - update w_i after each example.

$$w_i := w_i + \eta \cdot \delta(e) \cdot X_i(e)$$

-2 has been absorbed by η



+ approaches solution faster

− does not converges.

- Stochastic: choose examples at random.
- Batched Gradient descent:- update of some number ne of examples
  - (i) starts with ne=1 afterwards ne increases until ne = |E|
  - (i) vary the size of η start with large value, decrease latter.

```
PROCEDURE LinearLearner ( E, eta )
 • E : set of examples, each of the form <X1, X2, X3, …, Y>
 • eta : learning rate.

   initialize w0,…,wn randomly
   REPEAT
     FOR EACH example e in E DO
       Ycap := ∑i  wi * Xi(e)
       delta := Y(e) - Ycap
       update := eta * delta
       FOR EACH wi DO
         wi := wi + update * Xi(e)
   UNTIL some stop criteria is true
   RETURN w0,…,wn
```

from regression to Classification.

$X_0, X_1, \ldots X_n \in \mathbb{R}$ $\quad Y \in \mathbb{R}$ ← regression

$Y \in \{0, 1\}$ ← classification

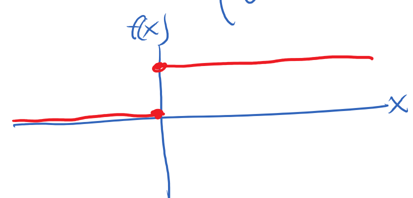
Examples

	Cats?	Celebrity?	Comedy?	Food?	Watch?
e1	True	False	False	True	Yes
e2	True	True	True	False	No
e3	False	False	True	False	Yes
e4	True	False	False	False	Yes
e5	True	False	True	False	No
e6	True	False	False	True	Yes
e7	False	False	True	True	Yes
e8	True	True	True	True	No
e9	T	F	F	T	?

regression : $\hat{Y} = \sum_{i=0}^{n} x_i W_i \qquad \hat{Y} \in \mathbb{R}$

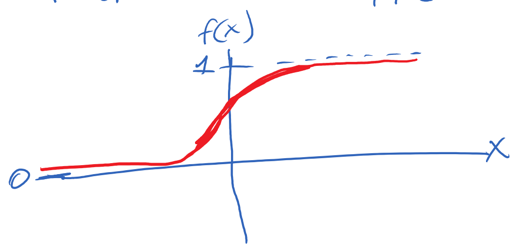
squash function $\hat{Y} = f\left(\sum_{i=0}^{n} x_i W_i\right)$ $\quad$ f squashes into $[0..1]$

one option for f: step function $f(x) \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise.} \end{cases}$



Second option for f: sigmoid function $f(x) = \frac{1}{1+e^{-x}} = sig(x)$

$sig'(x) = sig(x) \cdot (1 - sig(x))$



form of linear classification : "Logistic Regression"

$$\hat{Y}(e) = sig\left(\sum_{i=0}^{n} w_i X_i(e)\right)$$

$$\text{Error}(E) = \sum_{e \in E} \left(Y(e) - \hat{Y}(e)\right)^2$$

$$\frac{\partial\, \text{Error}(E)}{\partial w_i} = \frac{\partial}{\partial w_i} \sum_{e \in E} \left(Y(e) - sig\left(\sum_{i=0}^{n} w_i X_i(e)\right)\right)^2$$

$$\frac{\partial\, Error(E)}{\partial w_i} = \frac{\partial}{\partial w_i} \sum_{e \in E} \left(Y(e) - sig\left(\sum_{i=0}^{n} w_i X_i(e) \right) \right)^2$$

$$= \sum_{e \in E} 2 \cdot \left(\frac{\partial}{\partial w_i} \left(Y(e) - sig\left(\sum_{i=0}^{n} w_i X_i(e) \right) \right) \right)$$

$$= \sum_{e \in E} 2 \cdot \left(Y(e) - sig\left(\sum_{i=0}^{n} w_i X_i(e) \right) \right) \cdot \frac{\partial}{\partial w_i} \left(sig\left(\sum_{i=0}^{n} w_i X_i(e) \right) \right)$$

$$\cdot\, sig\left(\sum_{i=0}^{n} w_i X_i \right) \left(1 - sig\left(\sum_{i=0}^{n} w_i X_i(e) \right) \right) \cdot \frac{\partial}{\partial w_i} \sum_{i=0}^{n} w_i X_i(e)$$

$$\cdot\, X_i(e)$$

$$p(e) = sig\left(\sum_{i=0}^{n} w_i X_i(e) \right) = \hat{Y}(e)$$

$$\delta(e) = Y(e) - \hat{Y}(e)$$

$$\frac{\partial\, Error(E)}{\partial w_i} = \sum_{e \in E} -2 \cdot \delta(e) \cdot p \cdot (1-p) \cdot X_i(e)$$

```
PROCEDURE LogisticRegression ( E, eta )
• E : set of examples, each of the form <X1, X2, X3, …, Y>
  ○ Y is in {0,1}
• eta : learning rate.

  initialize w0,…,wn randomly
  REPEAT
    FOR EACH example e in E DO
      p := sig( ∑i  wi * Xi(e) )
      delta := Y(e) - p
      update := eta * delta * p * (1 - p)
      FOR EACH wi DO
        wi := wi + update * Xi(e)
  UNTIL some stop criteria is true
  RETURN w0,…,wn
```

	Cats?	Celebrity?	Comedy?	Food?	Watch?
e1	True	False	False	True	Yes
e2	True	True	True	False	No
e3	False	False	True	False	Yes
e4	True	False	False	False	Yes
e5	True	False	True	False	No
e6	True	False	False	True	Yes
e7	False	False	True	True	Yes
e8	True	True	True	True	No

$$\hat{Y}(e) = sig\left(w_0 + w_1\, Cats(e) + w_2\, Celebrity(e) + w_3\, Comedy(e) + w_4\, Food(e) \right)$$

Example (2):

$\langle X_1, X_2, X_3 .. X_n \rangle$ $Y = \{cat, dog, donut\}$

technicke.

$Y_{cat} = \{T, F\}$ indicator variables

$Y_{dog} = \{T, F\}$

$Y_{donut} = \{T, F\}$

Step function: $f(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise.} \end{cases}$

'60 

threshold.

Frank Rosenblatt : "Perceptron"

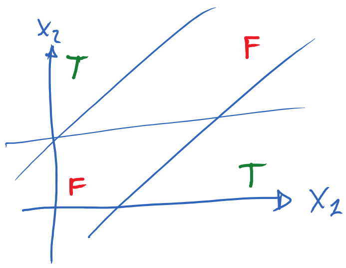
circuit $f(\sum w_i X_i)$

'58 "(Perceptrons) are the embryo of an electronic computer that will be able to talk, see, write, translate languages and reproduce itself and be conscious of its existence."

'69 = Marvin Minsky "Perceptrons"
Seymour Papert

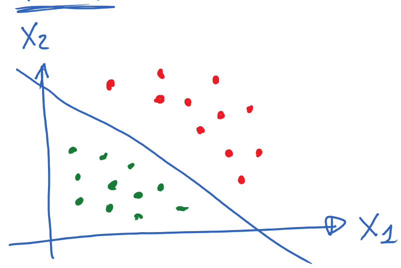
xor function

= first AI winter.



xor.

$\hat{Y}(E)$ is a line. (hyperplane)



Data non-linearly separatable

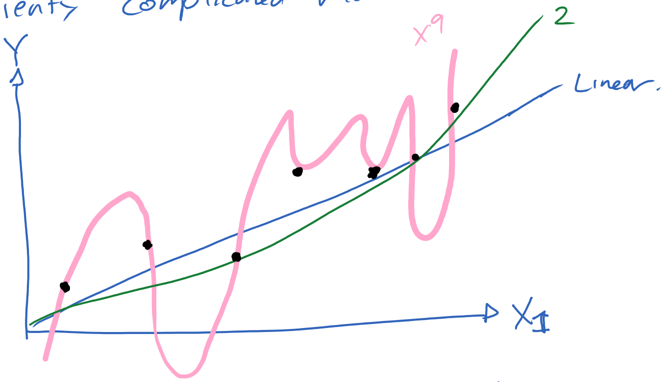
---

why just $\sum w_i X_i$???

a sufficiently complicated model can fit any data.
Y

x^9 , $/^2$

a Sufficiently complicated model can fit any data.



Ockham's Razor: when faced with more than one explanation prefer the simplest one.

modify algorithm with a "regularizer"
a component that rewards simplicity and punishes complexity

$$\hat{Y}(e) = sig\left(\sum_{i=0}^{n} w_i X_i\right) \qquad Error(E) = \sum_e \left(Y(e) - \hat{Y}(e)\right) + \lambda \underbrace{\left(\sum_{i=0}^{n} |w_i|\right)}_{\text{regularizer } L_1}$$

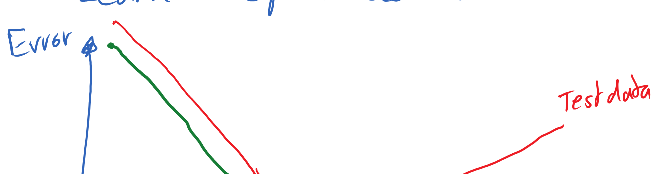
regularization parameter

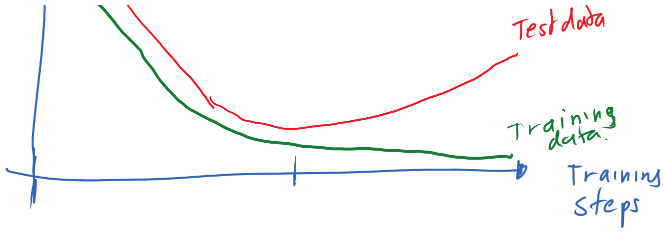
$$+ \lambda \left(\sum_{i=0}^{n} w_i^2\right)$$

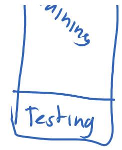
regularizer L_2.

$$\frac{\partial}{\partial w} = \left(Y(e) - \hat{Y}(e)\right)^2 \qquad \frac{\partial}{\partial w} f(g(x)) = f'(g(x)) \cdot \frac{\partial}{\partial w} g(x)$$

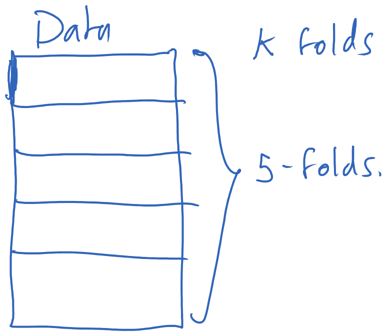
- Errors & Pitfalls
  - Bias: Learner to find imperfect model
    - representation model not good enough
    - Search is not good enough
  - Data is not good
    - Lack data
    - Data is noisy
  - Overfitting:
    Learner specializes to the training data.







One technique to avoid overfitting: <u>Cross-Validation</u>



pick 1 fold to test
train on remaining folds
Repeat K times.

Use <u>all</u> the data to train