EVENT DETECTION, MACHINE LEARNING AND THE MARITIME INDUSTRY

Florida State University, Department of Mathematics
Sept. 25^{th}, 2017

Pierre Garreau

Research associate ★ Dept. of Mathematics, Goethe University
CTO ★ Maritime Data Systems, Hamburg, Germany.

A WORD ON ANALYTICS AND FINANCE

Background

- Ph.D. thesis: Lévy processes, Copulas, SEM.
- Citi: Prepayment, default probabilities $2\bar{M}$ loans.
- Deutsche Bank: Optimization of Counterparty Exposure worldwide.
- Start-ups: Machine learning everywhere.

A WORD ON ANALYTICS AND FINANCE

Background

- Ph.D. thesis: Lévy processes, Copulas, SEM.
- Citi: Prepayment, default probabilities $2\bar{M}$ loans.
- Deutsche Bank: Optimization of Counterparty Exposure worldwide.
- Start-ups: Machine learning everywhere.

New Breed of Super Quants at NYU Prep for Wall Street

New York University is stepping into the breach by starting a Ph.D. program in data science in September to shape the emerging discipline. [...] MIT is gearing up a Ph.D. that includes data science and Harvard plans to jump into the field with a master's program in 2018.

CONTENTS

1· AIS & MARITIME, STATE MACHINES

RAW AIS DATA


```
!AIVDM,1,1„A,13P;tBwP0SOvhQ‘Nh4M9Owv40<1J,0*36,1505938796
!AIVDM,1,1„A,13P;tBwP0SOvhQ‘Nh4M9Owv40>‘<,0*13,1505938796
!AIVDM,1,1„A,14Sd9f001qOwhrrNbvrcjIR>0<1s,0*11,1505938796
!AIVDM,1,1„B,14Sd9f01iqOwhe‘NbwRsjaRT00Rm,0*4C,1505938796
!AIVDM,1,1„A,14Sd9f001qOwhLBNc0HKkqRp0>‘<,0*21,1505938796
!AIVDM,1,1„B,14Sd9f0viqOwh?NNc10cl9S<08Gb,0*6F,1505938796
```

PARSED AIS DATA

Table: Parsed AIS data

timestamp	cog	hdg	lat	lon	mmsi	sog
1505938796	160	150	51.502	-2.732	372204000	1.799
1505938796	131	151	51.503	-2.734	232003492	1.799
1505938796	179	145	51.503	-2.734	232003492	2.099
1505938826	162	150	51.501	-2.732	372204000	1.799
1505938856	162	149	51.501	-2.732	372204000	1.899

ASSISTED MOVEMENT

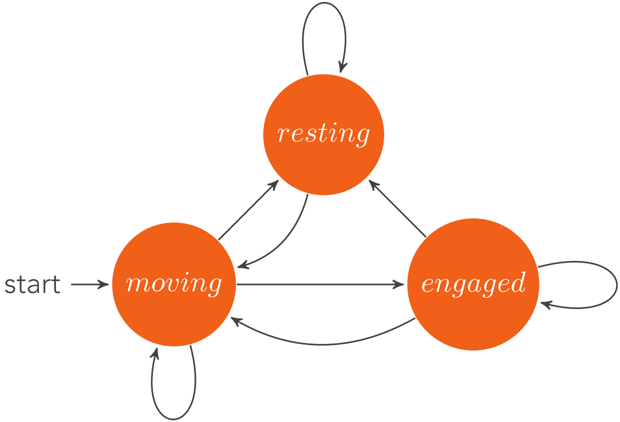
Figure: What do Tug boats do ?

ASSISTED MOVEMENTS

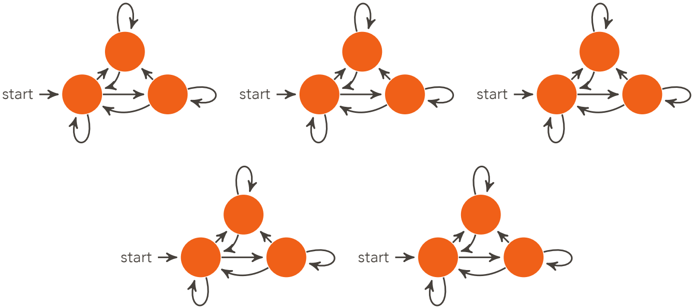
Table: SEAHAKE, Sea to Immingham Oil Terminal, Sep 21st 2017

Tug Name	Mob	Engaged	Disengaged	Rest
SVITZER KATHLEEN	07:19	07:28	08:23	08:33
SVITZER VALIANT	06:38	07:04	08:27	08:41

MODEL FOR VESSELS



MODEL FOR THE PORT



2· NEURAL NETWORKS

A BASIC CLASSIFIER

- Given *labelled* data $(x, y) = \{(x^{(m)}, y^{(m)})\}_{m=1}$, with $x^{(m)} \in \mathbb{R}^d$ and $y^{(m)} \in \{0, 1\}$, find a representation h such that
$$y = h(x; \theta, b) = \mathbb{P}(y = 1 | x; \theta, b).$$
- We postulate a model where $\theta \in \mathbb{R}^d, b \in \mathbb{R}$:
$$h(x; \theta, b) = g(\theta^t x + b) = \frac{1}{1 + \exp\{-\theta^t x - b\}}.$$
- Note that $g'(z) = g(z)(1 - g(z)) = a(1 - a)$.

A BASIC CLASSIFIER

- Given *labelled* data $(x, y) = \{(x^{(m)}, y^{(m)})\}_{m=1}$, with $x^{(m)} \in \mathbb{R}^d$ and $y^{(m)} \in \{0, 1\}$, find a representation h such that
$$y = h(x; \theta, b) = \mathbb{P}(y = 1|x; \theta, b).$$
- We postulate a model where $\theta \in \mathbb{R}^d, b \in \mathbb{R}$:
$$h(x; \theta, b) = g(\theta^t x + b) = \frac{1}{1 + \exp\{-\theta^t x - b\}.}$$
- Note that $g'(z) = g(z)(1 - g(z)) = a(1 - a)$.

A BASIC CLASSIFIER

- Given *labelled* data $(x, y) = \{(x^{(m)}, y^{(m)})\}_{m=1}$, with $x^{(m)} \in \mathbb{R}^d$ and $y^{(m)} \in \{0, 1\}$, find a representation h such that

$$y = h(x; \theta, b) = \mathbb{P}(y = 1 | x; \theta, b).$$

- We postulate a model where $\theta \in \mathbb{R}^d, b \in \mathbb{R}$:

$$h(x; \theta, b) = g(\theta^t x + b) = \frac{1}{1 + \exp\{-\theta^t x - b\}.}$$

- Note that $g'(z) = g(z)(1 - g(z)) = a(1 - a)$.

LIKELIHOOD, CROSS ENTROPY

- Find θ, b so as to maximize the *likelihood* of observing (x, y):

$$\mathcal{L}(\theta, b) = \prod_{m=1}^{M} f(y^{(m)} | x^{(m)}; \theta, b)$$

- Find θ, b so as to minimize the *cross entropy* $-\log \mathcal{L}(\theta, b)$ of observing (x, y) given θ, b:

$$\begin{aligned} E(\theta, b) = & - \frac{1}{m} \sum_{m=1}^{M} h(x^{(m)}; \theta, b) \log y^{(m)} \\ & + (1 - h(x^{(m)}; \theta, b)) \log(1 - y^{(m)}) \end{aligned}$$

- Solve:

$$\min_{\theta, b} J(\theta, b) = \min_{\theta, b} E(\theta, b) + \lambda ||\theta||^2$$

LIKELIHOOD, CROSS ENTROPY

- Find θ, b so as to maximize the *likelihood* of observing (x, y):

$$\mathcal{L}(\theta, b) = \prod_{m=1}^{M} f(y^{(m)} | x^{(m)}; \theta, b)$$

- Find θ, b so as to minimize the *cross entropy* $-\log \mathcal{L}(\theta, b)$ of observing (x, y) given θ, b:

$$\begin{aligned} E(\theta, b) = \quad & - \quad \frac{1}{m} \sum_{m=1}^{M} h(x^{(m)}; \theta, b) \log y^{(m)} \\ & + \quad (1 - h(x^{(m)}; \theta, b)) \log(1 - y^{(m)}) \end{aligned}$$

- Solve:

$$\min_{\theta, b} J(\theta, b) = \min_{\theta, b} E(\theta, b) + \lambda ||\theta||^2$$

LIKELIHOOD, CROSS ENTROPY

- Find θ, b so as to maximize the *likelihood* of observing (x, y):

$$\mathcal{L}(\theta, b) = \prod_{m=1}^{M} f(y^{(m)} | x^{(m)}; \theta, b)$$

- Find θ, b so as to minimize the *cross entropy* $-\log \mathcal{L}(\theta, b)$ of observing (x, y) given θ, b:

$$\begin{aligned} E(\theta, b) = \quad & - \quad \frac{1}{m} \sum_{m=1}^{M} h(x^{(m)}; \theta, b) \log y^{(m)} \\ & + \quad (1 - h(x^{(m)}; \theta, b)) \log(1 - y^{(m)}) \end{aligned}$$

- Solve:

$$\min_{\theta, b} J(\theta, b) = \min_{\theta, b} E(\theta, b) + \lambda ||\theta||^2$$

GRADIENT DESCENT

Algorithm 1: Gradient descent

Input: $\alpha, \epsilon, (x, y)$

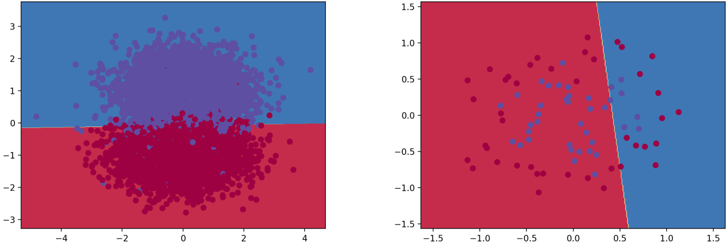
while $\triangle\theta > \epsilon$ *and* $\triangle b > \epsilon$ **do**

$\quad \theta := \theta - \alpha \nabla_\theta J(\theta, b)$

$\quad b := b - \alpha \nabla_b J(\theta, b)$

Output: θ, b

CLASSIFICATION BOUNDARY

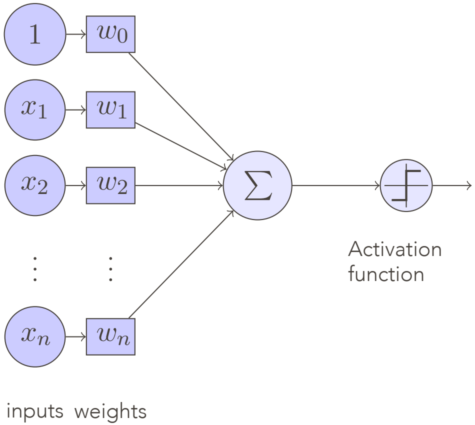


(a) Linearly separable

(b) Circular

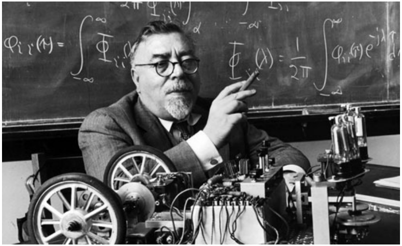
Figure: Logistic regression classifier on 2 different datasets

PERCEPTRON

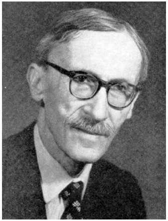

1
w_0
x_1
w_1
x_2
w_2
x_n
w_n
Σ
Activation function
inputs
weights

BRIEF HISTORY OF NEURAL NETWORKS

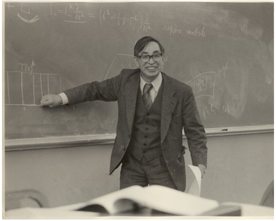
(a) Bachelier



(b) Wiener

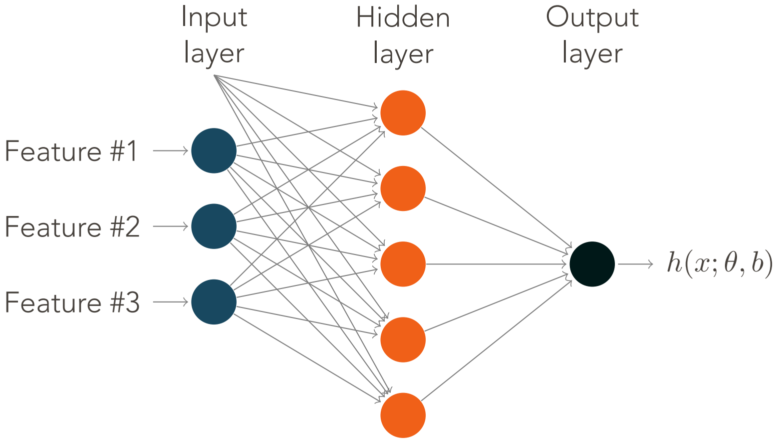


(c) Lévy



(d) Itô

3 LAYER FEED FORWARD NEURAL NETWORK



$$a^{(1)} = x$$

$$\begin{aligned} z^{(2)} &= \theta^{(1)} a^{(1)} + b^{(1)} \\ a^{(2)} &= g(z^{(2)}) \end{aligned}$$

FEEDFORWARD

Algorithm 2: NN 3 layers, feedforward

Input: θ, b, x

```
a^(1) = x
for l ← 1 to 2 do
    z^(l+1) = θ^(l) a^(l) + b^(l)
    a^(l+1) = g(z^(l+1))
y = a^(L)
```

$a^{(1)} = x$

for $l \leftarrow 1$ **to** 2 **do**

$\quad z^{(l+1)} = \theta^{(l)} a^{(l)} + b^{(l)}$

$\quad a^{(l+1)} = g(z^{(l+1)})$

$y = a^{(L)}$

Output: $y, a^{(2)}, a^{(3)}$

VALUE FUNCTION

- Find θ, b so as to minimize the *cross entropy* $-\log \mathcal{L}(\theta, b)$ of observing (x, y) given θ, b:

$$E_m(\theta, b) = -\frac{1}{2}||y^{(m)} - h(x^{(m)}; \theta, b)||^2$$

- Solve:

$$\min_{\theta,b} J(\theta, b) = \min_{\theta,b} \frac{1}{2m} \sum_{m=1}^{M} E_m(\theta, b) + \frac{1}{2}\lambda \sum_{l=1}^{L} ||\theta^{(l)}||^2$$

VALUE FUNCTION

- Find θ, b so as to minimize the *cross entropy* $-\log \mathcal{L}(\theta, b)$ of observing (x, y) given θ, b:

$$E_m(\theta, b) \quad = \quad -\frac{1}{2}||y^{(m)} - h(x^{(m)}; \theta, b)||^2$$

- Solve:

$$\min_{\theta,b} J(\theta, b) = \min_{\theta,b} \frac{1}{2m} \sum_{m=1}^{M} E_m(\theta, b) + \frac{1}{2}\lambda \sum_{l=1}^{L} ||\theta^{(l)}||^2$$

BACKPROPAGATION

The *backpropagation* algorithm comes from an application of the chain rule.

- The output layer $\theta^{(2)} \in \mathbb{R}^{1\times(K+1)}$ gives us

$$\begin{aligned}\frac{\partial E}{\partial \theta^{(2)}_{1k}} &= \frac{\partial E}{\partial z^{(3)}_1} \cdot \frac{\partial z^{(3)}_1}{\partial \theta^{(2)}_{1k}} \\ &= \delta^{(3)}_1 \cdot a^{(2)}_k\end{aligned}$$

- The hidden layer $\theta^{(1)} \in \mathbb{R}^{K\times(d+1)}$

$$\begin{aligned}\frac{\partial E}{\partial \theta^{(1)}_{kj}} &= \frac{\partial E}{\partial z^{(3)}_1} \cdot \frac{\partial z^{(3)}_1}{\partial z^{(2)}_k} \cdot \frac{\partial z^{(2)}_k}{\partial \theta^{(1)}_{kj}} \\ &= \delta^{(3)}_1 \cdot \theta^{(2)}_{1k} g'(z^{(2)}_k) \cdot a^{(1)}_j\end{aligned}$$

FEEDFORWARD, BACKPROPAGATION

Algorithm 3: NN 3 layers, backpropagation

Input: $a^{(2)}, a^{(3)}, \hat{y}$

$\delta^{(3)} = \partial E / \partial z_1^{(3)} = -(\hat{y} - a^{(3)}) g'(z^{(3)})$

for $l \leftarrow 2$ **to** 1 **do**

$\quad \delta_k^{(l)} = g'(z_k^{(l)}) \sum_{i=1}^{n_l} \theta_{ik}^{(l)} \delta_i^{(l+1)}$

$\quad \partial E / \partial \theta_{kj}^{(l)} = \delta_k^{(l+1)} a_j^{(l)}$

Output: θ, b

FEEDFORWARD NN ALGO

Algorithm 4: Gradient descent

Input: $\alpha, \epsilon, (x, y)$

```
while △θ > ϵ and △b > ϵ do
    ŷ := feedforward(x, θ, b)
    θ, b := backpropagation(x, y, ŷ)
```

Output: θ, b

CLASSIFICATION BOUNDARY

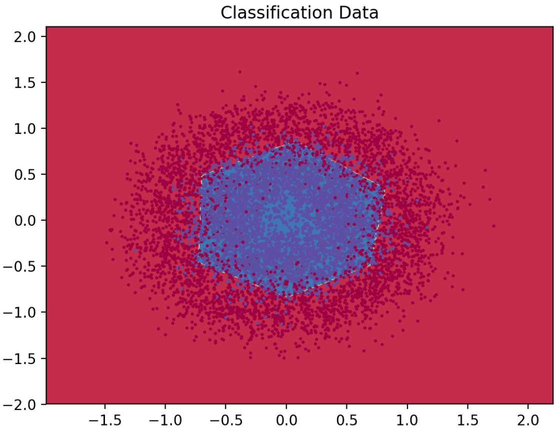


Figure: FF Neural Net, $3/3/1$, circular dataset

3. CLOSING

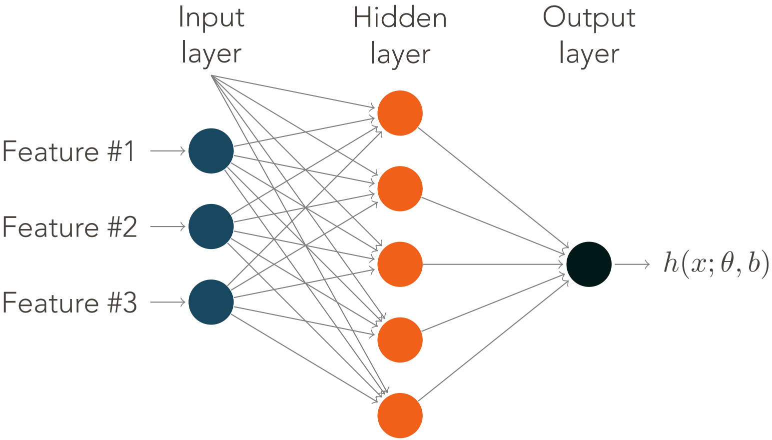
WHEN IT WORKS

Video of assisted movement.

WHEN IT DOES NOT WORK

Video of failed assisted movement.

RECURRENT NEURAL NETWORK



Thank You !