Data-driven Stroke Rehabilitation Using Machine Learning

Carlos Fernandez-Granda  
(NYU Courant Institute and Center for Data Science)

Joint work with Aakash Kaku (NYU Center for Data Science), Avinash Parnandi, Heidi Schambra, Anita Venkatesan  
(NYU School of Medicine)

Work supported by NIH R01 LM013316
Stroke and the brain

Ischemic Stroke

Cholesterol plaque buildup

Blood clot blocks artery
Post-stroke recovery

Pre-stroke function

Recovery

Time after stroke
Stroke is the **LEADING CAUSE** of disability in the US

Number of strokes is increasing

Broderick 2004 Stroke
Stroke and the arm

Stroke affects the arm in 77% of patients.

At 6 months, motor impairment still limits activities in 80% of patients.
Approaches to improving recovery

Pre-stroke function

Rehab training

Recovery

Time after stroke
Primary role of upper extremities in humans: act on objects

Functional UE movements have 4 components:
1. Reach
2. Contact (grasp/touch)
3. Object-specific action
4. Release
How much should we be dosing our patients?

Animals require 200-1000 reps/day to recover

Bell 2015 NNR

MacLellan 2011 NNR
What actually happens in stroke rehab?

Functional arm movements
- In only 51% of therapy sessions
- In these, 32 reps/session (range 1-420)

We are likely under-dosing our patients by 10x

Lang 2009 APMR
Measuring training dose is tough

Time

Motion capture systems

Activity monitors
Automatic identification

3D motion capture with wearable sensors
+
Machine learning
Automatic identification

3D linear velocity
3D angular acceleration
3 quaternion vectors
joint angles

reach
transport
Motion capture: inertial measurement units (IMUs)

- 3D linear and angular motion
- High precision
- Easy to wear
- No occlusion

- 100 Hz capture
- Record up to 10 h
- Transmit up to 150 m
Motion capture in reality
Machine learning

1. Gather data
2. Label the data
3. Build a model mapping data to labels
4. Apply the model on new data
Gather data
Data
Label data

“rest”  “reach”  “transport”

1 min video = 1 hour labeling
Build a model

A model is a function that maps data to labels

How do we build the function?

*Change it until the error is small*
Toy example

Data: Real numbers

Labels: 0 or 1
Model

Logistic function shifted by $t$
Model

Logistic function shifted by $t$
Problem: How to choose $t$?

$t = 0$, error: 1.499
Problem: How to choose t?

t = 25, error: 1.035
Error as a function of $t$
How to get to minimum

In real problems, we need to fit many parameters (not just one)

Data are multidimensional (not 1D)

Computing error for all possible values of the data is impossible!
How can we make progress?
Descent method

Move in the direction opposite to the derivative until it’s zero

New t = old t – constant * derivative at t

Constant controls how fast we move
Step 0
Step 0

t = 0.0, error: 1.499
Step 1
Step 1

t = 0.3, error: 1.492
Step 2
Step 2

t = 0.5, error: 1.485
Step 5

$t = 1.4$, error: 1.459
Step 10
Step 10

t = 5.9, error: 0.978
Step 15
Step 15

$t = 10.7$, error: 0.478
Step 25
Step 25

t = 15.4, error: 0.042