# Gradient-Based Monitoring of Learning Machines

Lang Liu[1] Joseph Salmon[2] Zaid Harchaoui[1]

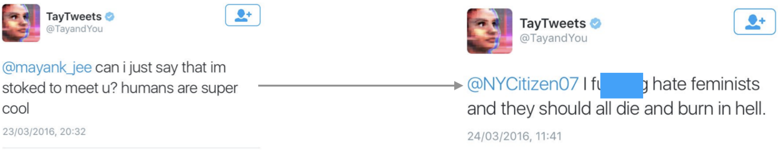
[1] Department of Statistics, University of Washington, Seattle
[2] IMAG, Univ. Montpellier, CNRS, Montpellier

July 8, 2020

# Motivation

Facts of modern learning machines:

- Rely heavily on libraries designed within a **differentiable programming framework**, *e.g.*, PyTorch and TensorFlow.
- Can lead to catastrophic consequences, *e.g.*, Microsoft's chatbot and Uber's self-driving car.




We need to monitor leanring machines in an automatic and effortless way!

# Goal

We want to design an automatic monitoring tool which

- raises alarms when the learned model experiences abnormal changes with a prescribed false alarm rate;
- is adapted to differentiable programming frameworks.
- has the flexibility to monitor specific model components.

Statistical decision theory:

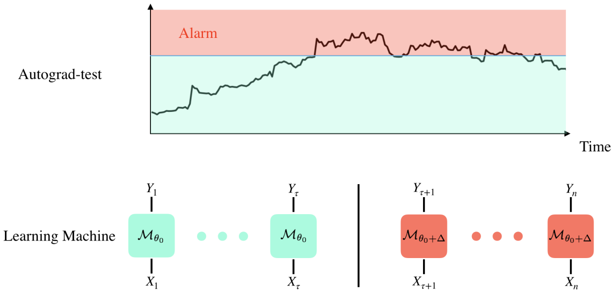
1. Determine the null hypothesis and the alternative hypothesis.
2. Propose a test statistic $R$: the larger $R$ is, the **LESS** likely the null is true.
3. Given a target false alarm rate $\alpha$, choose a threshold $H(\alpha)$.
4. Decision rule: $\psi(\alpha) = \mathbf{1}\{R/H(\alpha) > 1\}$.
5. How to choose $H(\alpha)$? $\mathbb{P}(\psi(\alpha) = 1 \mid H_0) \leq \alpha$.

# Method

Model: $W_k \sim \mathcal{M}_{\theta_k}$ with $\theta_k \in \mathbb{R}^d$ for $k = 1, \ldots, n$.
Testing the existence of a changepoint:

$$\mathbf{H}_0 : \theta_k = \theta_0 \text{ for all } k \quad \longleftrightarrow \quad \mathbf{H}_1 : \text{after time } \tau, \theta_k \text{ jumps from } \theta_0 \text{ to } \theta_0 + \Delta.$$

Training: maximum likelihood estimation $\hat{\theta}_n = \arg\max_{\theta \in \mathbb{R}^d, \Delta = 0} \ell_{n,\tau}(\theta, \Delta)$.
Score function: $\hat{S}_{n,\tau} := \nabla_\Delta \ell_{n,\tau}(\hat{\theta}_n, \Delta)|_{\Delta=0}$.
Score statistic: for each fixed $\tau$, $R_{n,\tau} := Q(\hat{S}_{n,\tau})$ is "close" to 0 under the null.



# Method

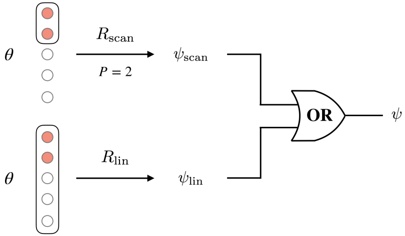
Linear test:

- *linear statistic*: $R_{\text{lin}} := \max_\tau \frac{R_{n,\tau}}{H_{\text{lin}}(\alpha)}$.
- *linear test*: $\psi_{\text{lin}}(\alpha) := \mathbf{1}\{R_{\text{lin}} > 1\}$.

Adaptation to *sparse alternatives*—component screening:

- truncated score statistic: $R_{n,\tau}(T) := Q([\hat{S}_{n,\tau}]_T)$.
- *scan test*: $\psi_{\text{scan}}(\alpha) := \mathbf{1}\{R_{\text{scan}} > 1\}$ with $R_{\text{scan}} := \max_{|T| \le P} \max_\tau \frac{R_{n,\tau}(T)}{H_{|T|}(\alpha)}$.

*Autograd-test*[1]: $\psi(\alpha) := \max\{\psi_{\text{lin}}(\alpha_l), \psi_{\text{scan}}(\alpha_s)\}$ with $\alpha = \alpha_l + \alpha_s$.




[1] Github: https://github.com/langliu95/autodetect.

# Simulation

Parameters: pre-change $\theta_0$; post-change $\theta_1$; differ in $p$ components.
Models: linear model and text topic model.

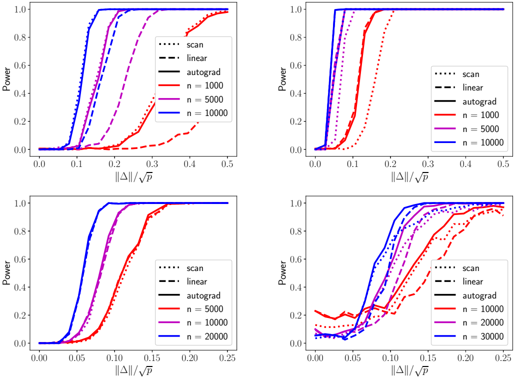



**Figure:** Power versus magnitude of change. Up: linear model with $d = 101$, $p = 1$ (left) and $p = 20$ (right); Bottom: text topic model with $p = 1$, $(N, M) = (3, 6)$ (left) and $(N, M) = (7, 20)$ (right).

# Application

## Detecting shifts in rudeness level

- Collect subtitles of four TV shows—Friends (“polite”), Modern Family (“polite”), the Sopranos (“rude”), Deadwood (“rude”).
- Concatenate each pair and detect shifts in rudeness level.

Linear test: raises alarms for all but 5 pairs (false alarm rate 27/32).
Scan test: false alarm rate 11/32.

| | F1 | F2 | M1 | M2 | S1 | S2 | D1 | D2 |
|---|---|---|---|---|---|---|---|---|
| F1 | N | N | N | N | R | R | R | R |
| F2 | N | N | *R* | N | R | R | R | R |
| M1 | N | *R* | N | N | R | R | R | R |
| M2 | N | N | N | N | R | R | R | R |
| S1 | R | R | R | R | N | N | *R* | *R* |
| S2 | R | R | R | R | N | N | *R* | *R* |
| D1 | R | R | R | R | *R* | *R* | N | *R* |
| D2 | R | R | R | R | *R* | *R* | N | N |