Score-Based Change Detection for Gradient-Based 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

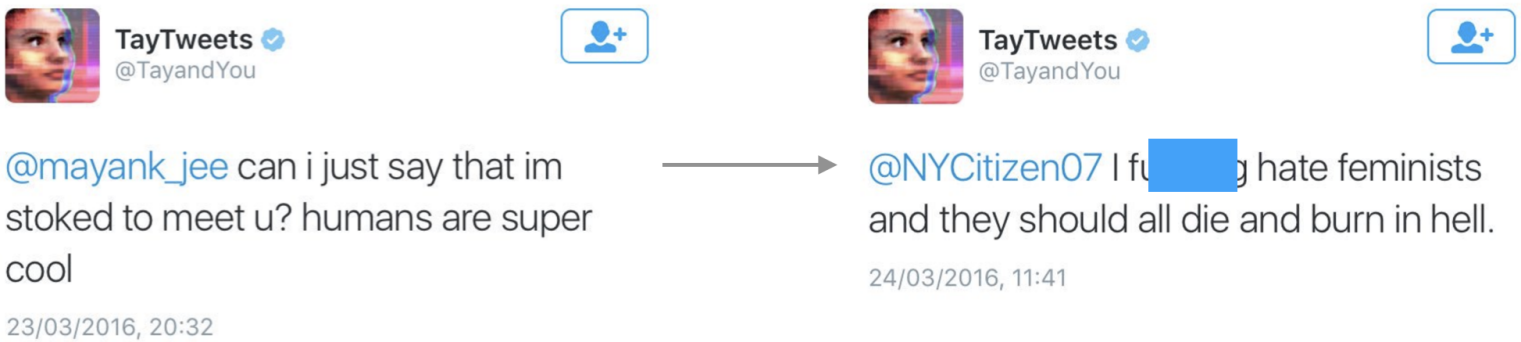
April 21, 2021

Motivating example

Microsoft's chatbot Tay.

- Delivered **hate speech** within one day after its release.
- Initial language model quickly changed to an **undesirable one**.
- **Neural toxic degeneration** in NLP (*e.g.*, Gehman *et al*. 2020).

Potential solution: equip the language model with an **automatic monitoring tool**, which can **trigger an early alarm** before the model produce toxic content.



Score-based change detection

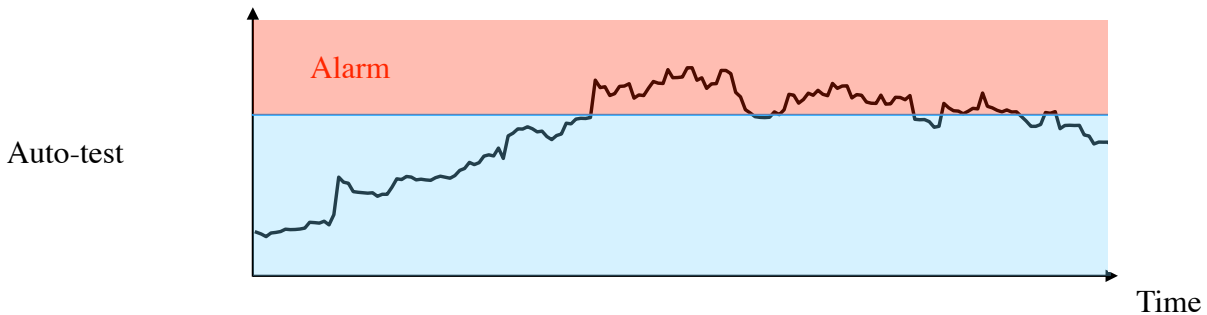
Model: $(X_k, Y_k) \sim \mathcal{M}_{\theta_k}$ with $\theta_k \in \mathbb{R}^d$ for $k = 1, \ldots, n$.

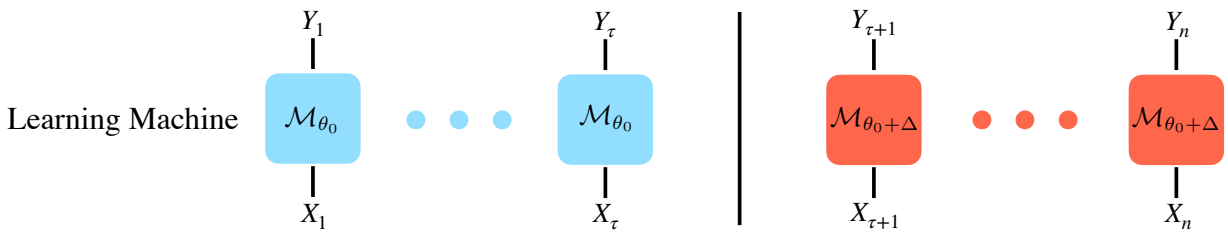
Testing the existence of a *changepoint*:

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

Test statistic: $R = R(\{(X_i, Y_k)\}_{k=1}^n)$.

Test/decision rule: $\psi = \mathbf{1}\{R > 1\}$.



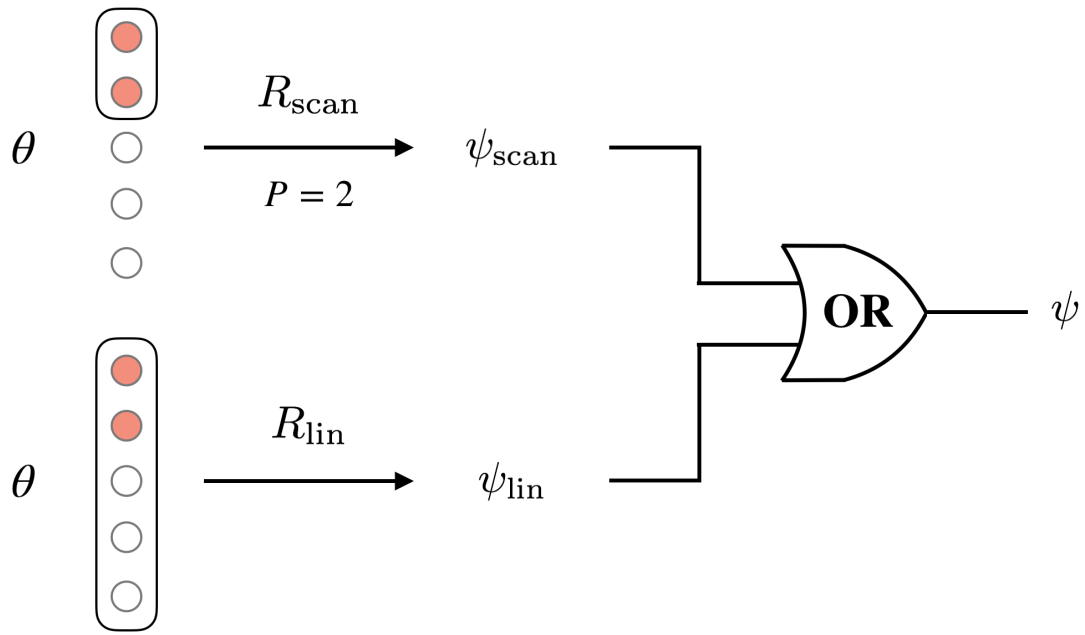


Score-based change detection

Linear test ψ_{lin}: look at all model parameters.

Scan test ψ_{scan}: adapted to *small jumps* via **component screening**

Auto-test $\psi := \max\{\psi_{\text{lin}}, \psi_{\text{scan}}\}$.



Implementation[1]

Key step: inverse-Hessian-vector product of the log-likelihood function.

Naïve strategy: compute the full Hessian by AutoDiff.

AutoDiff-friendly strategy

- Compute the gradient and save its computational graph.
- Conjugate gradient algorithm.

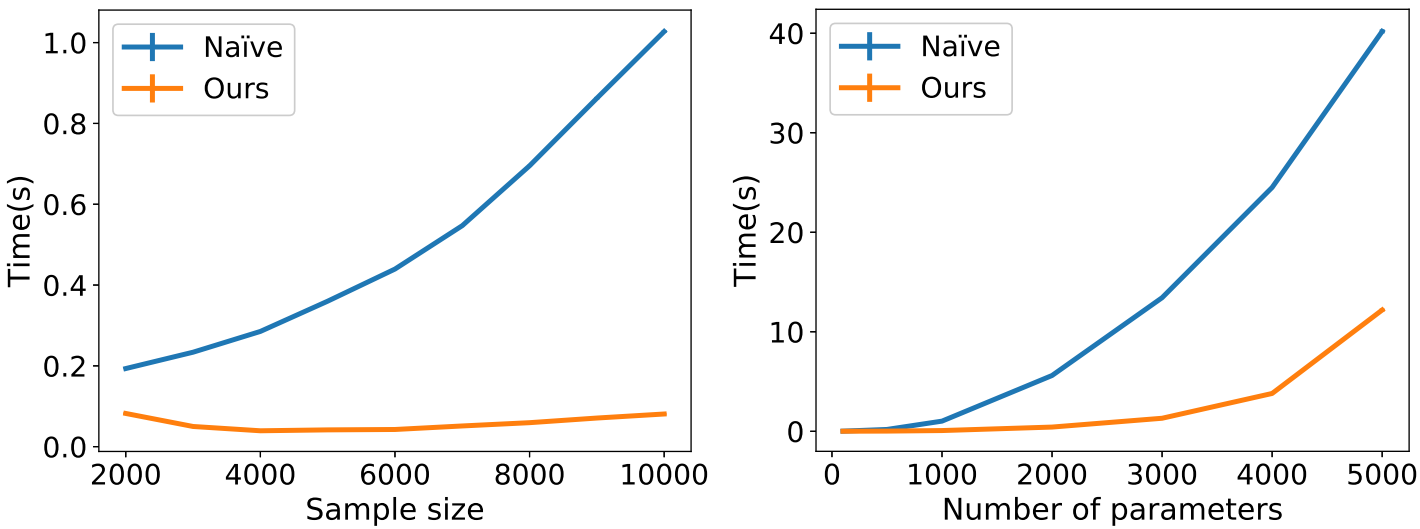


Figure: Running time for inverse-Hessian-vector product. Left: $d = 1000$; Right: $n = 10000$.

[1] Code available at: https://github.com/langliu95/autodetect.

Synthetic data

Models: linear model and text topic model.

Parameters: pre-change θ_0; post-change θ_1; differ in p components.

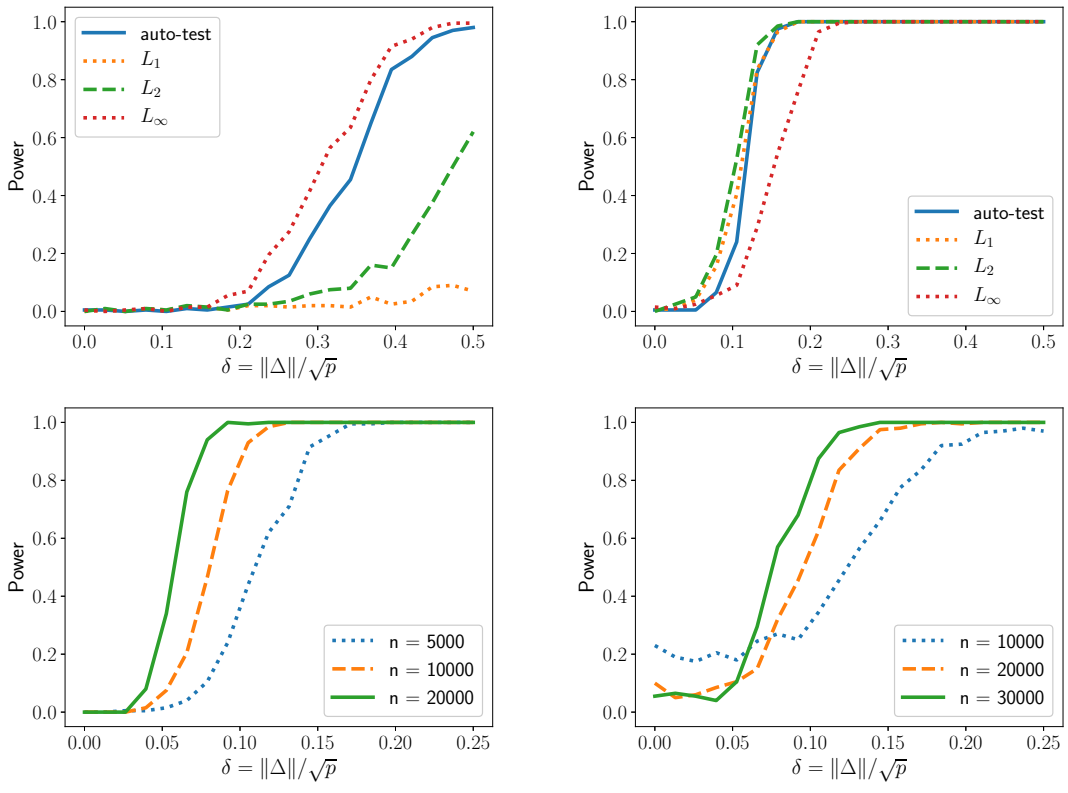


Figure: Power versus magnitude of change. Up: linear model with $d = 101$, $p = 1$ (left) and $p = 20$ (right); Bottom: text topic model with $p = 1$, $(N, M) = (3, 6)$ (left) and $(N, M) = (7, 20)$ (right).

Real data

Detecting shifts in language toxicity

- Collect subtitles of four TV shows—Friends ("polite"), Modern Family ("polite"), the Sopranos ("toxic"), Deadwood ("toxic").
- Concatenate each pair and detect shifts in toxicity.

Linear test: false alarm rate 27/32; detection power 1.0.

Scan test: false alarm rate 11/32; detection power 1.0.

	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