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, University of Montpellier, CNRS, Montpellier

ABSTRACT

The widespread use of machine learning algorithms calls for automatic change detection algorithms to monitor their behavior over time. As a machine learning algorithm learns from a continuous, possibly evolving, stream of data, it is desirable and often critical to supplement it with a companion change detection algorithm to facilitate its monitoring and control. We present a generic score-based change detection method that can detect a change in any number of components of a machine learning model trained via empirical risk minimization. This proposed statistical hypothesis test can be readily implemented for such models designed within a differentiable programming framework. We establish the consistency of the hypothesis test and show how to calibrate it to achieve a prescribed false alarm rate. We illustrate the versatility of the approach on synthetic and real data.

***Index Terms*—** Change detection, differentiable programming, system monitoring.

1. INTRODUCTION

Statistical machine learning models are fostering progress in numerous technological applications, *e.g.,* visual object recognition and language processing, as well as in many scientific domains, *e.g.,* genomics and neuroscience. This progress has been fueled recently by statistical machine learning libraries designed within a differentiable programming framework such as PyTorch [1] and TensorFlow [2].

Gradient-based optimization algorithms such as accelerated batch gradient methods are then well adapted to this framework, opening up the possibility of gradient-based training of machine learning models from a continuous stream of data. As a learning system learns from a continuous, possibly evolving, data stream, it is desirable to supplement it with tools facilitating its monitoring in order to prevent the learned model from experiencing abnormal changes.

Recent remarkable failures of intelligent learning systems such as Microsoft's chatbot [3] and Uber's self-driving car [4] show the importance of such tools. In the former case, the initially learned language model quickly changed to an undesirable one, as it was being fed data through interactions with users. The addition of an automatic monitoring tool can potentially prevent a debacle by triggering an early alarm, drawing the attention of its designers and engineers to an abnormal change of a language model.

To keep up with modern learning machines, the monitoring of machine learning models should be automatic and effortless in the same way that the training of these models is now automatic and effortless. Humans monitoring machines should have at hand automatic monitoring tools to scrutinize a learned model as it evolves over time. Recent research in this area is relatively limited.

We introduce a generic change monitoring method called *auto-test* based on statistical decision theory. This approach is aligned with machine learning libraries developed in a differentiable programming framework, allowing us to seamlessly apply it to a large class of models implemented in such frameworks. Moreover, this method is equipped with a *scanning* procedure, enabling it to detect *small jumps* occurring on an unknown subset of model parameters. The proofs and more details can be found in [5]. The code is publicly available at *github.com/langliu95/autodetect*.

Previous work on change detection. Change detection is a classical topic in statistics and signal processing; see [6, 7] for a survey. It has been considered either in the offline setting, where we test the null hypothesis with a prescribed false alarm rate, or in the online setting, where we detect a change as quickly as possible. Depending on the type of change, the change detection problem can be classified into two main categories: change in the model parameters [8, 9] and change in the distribution of data streams [10, 11, 12]. We focus on testing the presence of a change in the model parameters.

Test statistics for detecting changes in model parameters are usually designed on a case-by-case basis; see [6, 13, 14, 15, 16] and references therein. These methods are usually based on (possibly generalized) likelihood ratios or on residuals and therefore not amenable to differentiable programming. Furthermore, these methods are limited to *large jumps*, *i.e.,* changes occurring simultaneously on all model parameters, in contrast to ours.

This work was supported by NSF DMS 2023166, DMS 1839371, DMS 1810975, CCF 2019844, CCF-1740551, CIFAR-LMB, and research awards.

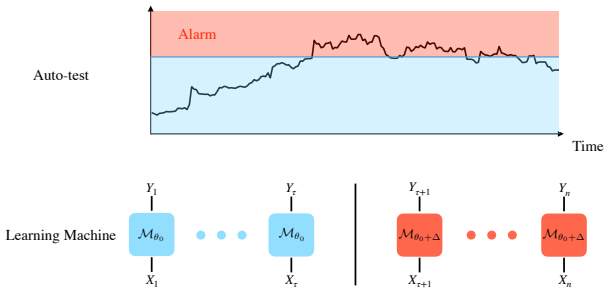

Fig. 1: Illustration of monitoring a learning machine.

2. SCORE-BASED CHANGE DETECTION

Let $W_{1:n} := \{W_k\}_{k=1}^n$ be a sequence of observations. Consider a family of machine learning models $\{\mathcal{M}_\theta : \theta \in \Theta \subset \mathbb{R}^d\}$ such that $W_k = \mathcal{M}_\theta(W_{1:k-1}) + \varepsilon_k$, where $\{\varepsilon_k\}_{k=1}^n$ are independent and identically distributed (*i.i.d.*) random noises. To learn this model from data, we choose a loss function L and estimate model parameters by solving the problem:

$$\hat{\theta}_n := \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{k=1}^{n} L\big(W_k, \mathcal{M}_\theta(W_{1:k-1})\big) \ .$$

This encompasses constrained empirical risk minimization (ERM) and constrained maximum likelihood estimation (MLE). For simplicity, we assume the model is *correctly specified*, *i.e.,* there exists a true value $\theta_0 \in \Theta$ from which the data are generated.

Under abnormal circumstances, this true value may not remain the same for all observations. Hence, we allow a potential parameter change in the model, that is, $\theta = \theta_k$ may evolve over time:

$$W_k = \mathcal{M}_{\theta_k}(W_{1:k-1}) + \varepsilon_k \ .$$

A time point $\tau \in [n-1] := \{1, \dots, n-1\}$ is called a *changepoint* if there exists $\Delta \neq 0$ such that $\theta_k = \theta_0$ for $k \leq \tau$ and $\theta_k = \theta_0 + \Delta$ for $k > \tau$. We say that there is a jump (or change) in the data sequence if such a changepoint exists. We aim to determine if there exists a jump in this sequence, which we formalize as a hypothesis testing problem.

(P0) Testing the presence of a jump

$\mathbf{H}_0 : \theta_k = \theta_0$ for all $k = 1, \dots, n$

$\mathbf{H}_1$: after some time τ, θ_k jumps from θ_0 to $\theta_0 + \Delta$.

We focus on models whose loss $L(W_k, \mathcal{M}_\theta(W_{1:k-1}))$ can be written as $-\log p_\theta(W_k|W_{1:k-1})$ for some conditional probability density p_θ. For instance, the squared loss function is associated with the negative log-likelihood of a Gaussian density; for more examples, see, *e.g.,* [17]. In the remainder of the paper, we will work with this probabilistic formulation for convenience, and we refer to the corresponding loss as the probabilistic loss.

Algorithm 1 Auto-test

1: **Input:** data $(W_i)_{i=1}^n$, log-likelihood ℓ, levels α_l and α_s, and maximum cardinality P.
2: **for** $\tau = 1$ **to** $n-1$ **do**
3: Compute $R_n(\tau)$ in (1) using AutoDiff.
4: Compute $R_n(\tau, P; \alpha)$ in (3).
5: **end for**
6: **Output:** $\psi_{\text{auto}}(\alpha) = \max\{\psi_{\text{lin}}(\alpha_l), \psi_{\text{scan}}(\alpha_s)\}$ in (4).

Remark. Discriminative models can also fit into this framework. Let $(X_1, Y_1), \dots, (X_n, Y_n)$ be *i.i.d.* observations, then the loss function reads $L(Y_k, \mathcal{M}_\theta(X_k))$. If, in addition, L is a probabilistic loss, then the associated conditional probability density is $p_\theta(Y_k|X_k)$.

2.1. Likelihood score and score-based testing.

Let $\mathbb{1}\{\cdot\}$ be the indicator function. Given $\tau \in [n-1]$ and $1 \leq s \leq t \leq n$, we define the conditional log-likelihood under the alternative as

$$\ell_{s:t}(\theta, \Delta; \tau) := \sum_{k=s}^{t} \log p_{\theta+\Delta\mathbb{1}\{k>\tau\}}(W_k|W_{1:k-1}) \ .$$

We will write $\ell_{s:t}(\theta, \Delta)$ for short if there is no confusion. Under the null, we denote by $\ell_{s:t}(\theta) := \ell_{s:t}(\theta, 0; n)$ the conditional log-likelihood. The *score function w.r.t.* θ is defined as $S_{s:t}(\theta) := \nabla_\theta \ell_{s:t}(\theta)$, and the *observed Fisher information w.r.t.* θ is denoted by $\mathcal{I}_{s:t}(\theta) := -\nabla_\theta^2 \ell_{s:t}(\theta)$.

Given a hypothesis testing problem, the first step is to propose a *test statistic* R_n such that the larger R_n is, the less likely the null hypothesis is true. Then, for a prescribed *significance level* $\alpha \in (0, 1)$, we calibrate this test statistic by a threshold $r_0 := r_0(\alpha)$, leading to a test $\mathbb{1}\{R_n > r_0\}$, *i.e.,* we reject the null if $R_n > r_0$. The threshold is chosen such that the *false alarm rate* or *type I error rate* is asymptotically controlled by α, *i.e.,* $\limsup_{n\to\infty} \mathbb{P}(R_n > r_0 \mid \mathbf{H}_0) \leq \alpha$. We say that such a test is *consistent in level*. Moreover, we want the *detection power*, *i.e.,* the conditional probability of rejecting the null given that it is false, to converge to 1 as n goes to infinity. And we say such a test is *consistent in power*.

Let us follow this procedure to design a test for Problem (P0). We start with the case when the changepoint τ is fixed. A standard choice is the *generalized score statistic* given by

$$R_n(\tau) := S_{\tau+1:n}^\top(\hat{\theta}_n)\mathcal{I}_n(\hat{\theta}_n; \tau)^{-1} S_{\tau+1:n}(\hat{\theta}_n) \ , \quad (1)$$

where $\mathcal{I}_n(\hat{\theta}_n; \tau)$ is the *partial observed information w.r.t.* Δ [18, Chapter 2.9] defined as

$$\mathcal{I}_{\tau+1:n}(\hat{\theta}_n) - \mathcal{I}_{\tau+1:n}(\hat{\theta}_n)^\top \mathcal{I}_{1:n}(\hat{\theta}_n)^{-1} \mathcal{I}_{\tau+1:n}(\hat{\theta}_n). \quad (2)$$

To adapt to an unknown changepoint τ, a natural statistic is $R_{\text{lin}} := \max_{\tau\in[n-1]} R_n(\tau)$. And, given a significance level

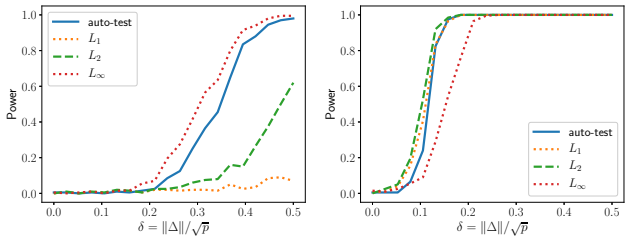

Fig. 2: Power curves for a linear model with $d = 101$ (left: $p = 1$; right: $p = 20$). The sample size is $n = 1000$.

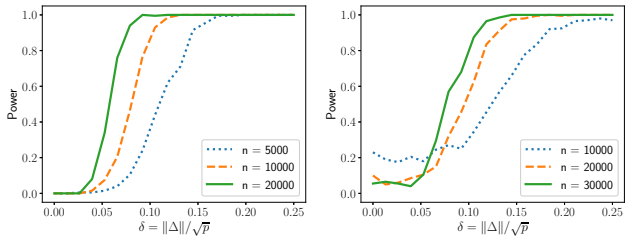

Fig. 3: Power curves of the *auto-test* for a text topic model with $p = 1$ (left: $(N, M) = (3, 6)$; right: $(N, M) = (7, 20)$).

α, the decision rule reads $\psi_{\text{lin}}(\alpha) := \mathbb{1}\{R_{\text{lin}} > H_{\text{lin}}(\alpha)\}$, where $H_{\text{lin}}(\alpha)$ is a prescribed threshold discussed in Sec. 3. We call R_{lin} the *linear statistic* and ψ_{lin} the *linear test*.

2.2. Sparse alternatives

There are cases when the jump only happens in a small subset of components of θ_0. The linear test, which is built assuming the jump is large, may fail to detect such small jumps. Therefore, we also consider *sparse alternatives*.

(P1) Testing the presence of a small jump:

$\mathbf{H}_0 : \theta_k = \theta_0$ for all $k = 1, \ldots, n$

$\mathbf{H}_1$: after some time τ, θ_k jumps from θ_0 to $\theta_0 + \Delta$, where Δ has at most P nonzero entries.

Here P is referred to as the *maximum cardinality*, which is set to be much smaller than d, the dimension of θ. We denote by T the changed components, *i.e.,* $\Delta_T \neq 0$ and $\Delta_{[d]\setminus T} = 0$.

Given a fixed T, we consider the *truncated statistic*

$$R_n(\tau, T) = S^{\top}_{\tau+1:n}(\hat{\theta}_n)_T \left[\mathcal{I}_n(\hat{\theta}_n; \tau)_{T,T}\right]^{-1} S_{\tau+1:n}(\hat{\theta}_n)_T.$$

Let $\mathcal{T}_p$ be the collection of all subsets of size p of $[d]$. To adapt to unknown T, we use

$$R_n(\tau, P; \alpha) := \max_{p\in[P]} \max_{T\in\mathcal{T}_p} H_p(\alpha)^{-1} R_n(\tau, T) \ , \qquad (3)$$

where we use a different threshold $H_p(\alpha)$ for each $p \in [P]$. Finally, since τ is also unknown, we propose $R_{\text{scan}}(\alpha) := \max_{\tau\in[n-1]} R_n(\tau, P; \alpha)$, with decision rule $\psi_{\text{scan}}(\alpha) := \mathbb{1}\{R_{\text{scan}}(\alpha) > 1\}$. We call $R_{\text{scan}}(\alpha)$ the *scan statistic* and ψ_{scan} the *scan test*.

To combine the respective strengths of these two tests, we consider the test

$$\psi_{\text{auto}}(\alpha) := \max\{\psi_{\text{lin}}(\alpha_l), \psi_{\text{scan}}(\alpha_s)\} \ , \qquad (4)$$

with $\alpha_l + \alpha_s = \alpha$, and we refer to it as the *auto-test*. The choice of α_l and α_s should be based on prior knowledge regarding how likely the jump is small. We illustrate how to monitor a learning machine with *auto-test* in Fig. 1.

2.3. Differentiable programming

An attractive feature of *auto-test* is that it can be computed by inverse-Hessian-vector products. That opens up the possibility to implement it easily using a machine learning library designed within a differentiable programming framework. Indeed, the inverse-Hessian-vector product can then be efficiently computed via automatic differentiation. The algorithm to compute the *auto-test* is presented in Alg. 1.

3. LEVEL AND POWER

We summarize the asymptotic behavior of the proposed score-based statistics under null and alternatives.

Proposition (Null hypothesis)**.** *Under the null hypothesis and certain conditions, we have, for any subset $T \subset [d]$ and $\tau_n \in \mathbb{N}$ such that $\tau_n/n \to \lambda \in (0, 1)$,*

$$R_n(\tau_n) \to_d \chi^2_d \quad \textit{and} \quad R_n(\tau_n, T) \to_d \chi^2_{|T|},$$

where we denote by $\to_d$ the convergence in distribution. In particular, with thresholds $H_{lin}(\alpha) = q_{\chi^2_d}(\alpha/n)$ and $H_p(\alpha) = q_{\chi^2_p}\big(\alpha/[\binom{d}{p} n(p+1)^2]\big)$, the tests $\psi_{lin}(\alpha)$, $\psi_{scan}(\alpha)$ and $\psi_{auto}(\alpha)$ are consistent in level, where $q_D(\alpha)$ is the upper α-quantile of the distribution D.

Most conditions in the above Proposition are standard. In fact, under suitable regularity conditions, they hold true for *i.i.d.* models, hidden Markov models [19, Chapter 12], and stationary autoregressive moving-average models [20, Chapter 13].

The next proposition verifies the consistency in power of the proposed tests under fixed alternatives.

Proposition (Fixed alternative hypothesis)**.** *Assume the observations are independent, and the alternative hypothesis is true with a fixed change parameter Δ. Let the changepoint τ_n be such that $\tau_n/n \to \lambda \in (0, 1)$. Under certain conditions, the tests $\psi_{lin}(\alpha)$, $\psi_{scan}(\alpha)$ and $\psi_{auto}(\alpha)$ are consistent in power.*

4. EXPERIMENTS

We apply our approach to detect changes on synthetic data and on real data. We summarize the settings and our findings.

Synthetic data. For each model, we generate the first half sample from the pre-change parameter θ_0 and generate the second half from the post-change parameter θ_1, where θ_1 is obtained by adding δ to the first p components of θ_0. Next, we run the proposed *auto-test* to monitor the learning process, where the significance levels are set to be $\alpha = 2\alpha_l = 2\alpha_s = 0.05$ and the maximum cardinality $P = \lfloor \sqrt{d} \rfloor$. We repeat this procedure 200 times and approximate the detection power by rejection frequency. Finally, we plot the power curves by varying δ. Note that the value at $\delta = 0$ is the empirical false alarm rate.

Additive model. We consider a linear model with 101 parameters and investigate two sparsity levels, $p = 1$ and $p = 20$. We compare the *auto-test* with three baselines given by the L_a norm of the score function for $a \in \{1, 2, \infty\}$, where these baselines are calibrated by the empirical quantiles of their limiting distributions. Note that the linear test corresponds to the L_2 norm with a proper normalization. And the scan test with $P = 1$ corresponds to the L_∞ norm. As shown in Fig. 2, when the change is sparse, *i.e.,* a small jump, the *auto-test* and L_∞ test have similar power curves and outperform the rest of the tests significantly. When the change is less sparse, *i.e.,* a large jump, all tests' performance gets improved, with the L_∞ test being less powerful than the other three. This empirically illustrates that (1) the L_∞ test work better in detecting sparse changes, (2) the L_1 test and the L_2 test are more powerful for non-sparse changes and (3) the *auto-test* achieves comparable performance in both situations.

The proposed *auto-test* is calibrated by its large sample properties and the Bonferroni correction. This strategy tends to result in tests that are too conservative, with empirical false alarm rates largely below 0.05. We also use resampling-based strategy to calibrate the *auto-test*, *i.e.,* generating bootstrap samples and calibrating the test using the quantiles of the test statistics evaluated on bootstrap samples. The empirical false alarm rates are around 0.065 for both $p = 1$ and $p = 20$.

Text topic model. We consider a text topic model [21] and investigate the *auto-test* for different sample sizes. This model is a hidden Markov model whose emission distribution has a special structure. We examine two parameter schemes: $(N, M) \in \{(3, 6), (7, 20)\}$, where N is the number of hidden states and M is the number of categories of the emission distribution, and p is set to be 1. As demonstrated in Fig. 3, for the first scheme, all tests have small false alarm rates, and their power rises as the sample size increases. For the second scheme, the false alarm rate is out of control in the beginning, but this problem is alleviated as the sample size increases. This empirically verifies that the *auto-test* is consistent in both level and power even for dependent data.

Real data. We collect subtitles of the first two seasons of four TV shows—Friends (F), Modern Family (M), the Sopranos (S) and Deadwood (D)—where the former two are viewed as "polite" and the latter two as "rude". For every pair of seasons, we concatenate them, and train the text topic model with $N = \lfloor \sqrt{n/100} \rfloor$ and M being the size of vocabulary built from the training corpus. The task is to detect changes in the rudeness level. As an analogy, the text topic model here corresponds to a chatbot, and subtitles are viewed as interactions with users. We want to know whether the conversation gets rude as the chatbot learns from the data.

Table 1: Decision of the scan test on the TV-show application: each (row, column) pair stands for a concatenation; "R" means reject and "N" means not reject. Red entries are false alarms.

	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

The linear test, *i.e.,* the *auto-test* with $\alpha_l = \alpha$ and $\alpha_s = 0$, does a perfect job in reporting shifts in rudeness level. However, it has a high false alarm rate $(27/32)$. This is expected since the linear test may capture the difference in other aspects, *e.g.,* topics of the conversation. The scan test, *i.e.,* the *auto-test* with $\alpha_l = 0$ and $\alpha_s = \alpha$, has much lower false alarm rate $(11/32)$. Moreover, as shown in Table 1, there are only two false alarms in the most interesting case, where the sequence starts with a polite show. We note that this problem is hard, since rudeness is not the only factor that contributes to the difference between two shows. The results are promising since we benefit from exploiting the sparsity even without knowing which model components are related to the rudeness level.

5. CONCLUSION

We introduced a change monitoring method called *auto-test* that is well suited to machine learning models implemented within a differentiable programming framework. The extension of this approach to penalized maximum likelihood or regularized empirical risk estimation in a high dimensional setting is an interesting venue for future work.

6. REFERENCES

[1] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga,

A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.

[2] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, Craig Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I.J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Józefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D.G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P.A. Tucker, V. Vanhoucke, V. Vasudevan, F.B. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, vol. abs/1603.04467, 2016.

[3] R. Metz, "Microsoft's neo-Nazi sexbot was a great lesson for makers of AI assistants," *Artificial Intelligence*, March 2018.

[4] W. Knight, "A self-driving Uber has killed a pedestrian in Arizona," *Ethical Tech*, March 2018.

[5] L. Liu, J. Salmon, and Z. Harchaoui, "Score-based change detection for gradient-based learning machines," *arXiv preprint*, 2021.

[6] M. Basseville and I.V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*, Prentice Hall, Inc., 1993.

[7] A. Tartakovsky, I. Nikiforov, and M. Basseville, *Sequential Analysis: Hypothesis Testing and Changepoint Detection*, Taylor & Francis, 2014.

[8] D.V. Hinkley, "Inference about the change-point in a sequence of random variables," *Biometrika*, vol. 57, no. 1, 1970.

[9] J. Deshayes and D. Picard, "Off-line statistical analysis of change-point models using non parametric and likelihood methods," in *Detection of Abrupt Changes in Signals and Dynamical Systems*. 1985, Springer.

[10] G. Lorden, "Procedures for reacting to a change in distribution," *The Annals of Mathematical Statistics*, vol. 42, no. 6, 1971.

[11] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *VLDB*, 2004.

[12] J.P. Cunningham, Z. Ghahramani, and C.E. Rasmussen, "Gaussian processes for time-marked time-series data," in *AISTATS*, 2012.

[13] E.G. Carlstein, H.G. Müller, and D. Siegmund, *Change-point problems*, Institute of Mathematical Statistics, 1994.

[14] Q. Zhang, M. Basseville, and A. Benveniste, "Early warning of slight changes in systems," *Automatica*, vol. 30, no. 1, 1994, Special issue on statistical signal processing and control.

[15] M. Csörgő and L. Horváth, *Limit Theorems in Change-Point Analysis*, Wiley Series in Probability and Statistics. Wiley, 1997.

[16] F. Enikeeva and Z. Harchaoui, "High-dimensional change-point detection under sparse alternatives," *Annals of Statistics*, vol. 47, no. 4, 2019.

[17] K.P. Murphy, *Machine learning: A Probabilistic Perspective*, MIT press, 2012.

[18] J. Wakefield, *Bayesian and Frequentist Regression Methods*, Mathematics and Statistics. Springer, 2013.

[19] P.J. Bickel, Y. Ritov, and T. Rydén, "Asymptotic normality of the maximum-likelihood estimator for general hidden markov models," *Annals of Statistics*, vol. 26, no. 4, 1998.

[20] R. Douc, E. Moulines, and D. Stoffer, *Nonlinear Time Series: Theory, Methods and Applications with R Examples*, Chapman and Hall/CRC, 2014.

[21] K. Stratos, M. Collins, and D. Hsu, "Model-based word embeddings from decompositions of count matrices," in *ACL-IJCNLP*, 2015, vol. 1.