Active Learning for Accurate Estimation of Linear Models

Carlos Riquelme [1] **Mohammad Ghavamzadeh** [2] **Alessandro Lazaric** [3]

Abstract

We explore the sequential decision-making problem where the goal is to estimate a number of linear models uniformly well, given a shared budget of random contexts independently sampled from a known distribution. For each incoming context, the decision-maker selects one of the linear models and receives an observation that is corrupted by the unknown noise level of that model. We present Trace-UCB, an adaptive allocation algorithm that learns the models' noise levels while balancing contexts accordingly across them, and prove bounds for its simple regret in both expectation and high-probability. We extend the algorithm and its bounds to the high dimensional setting, where the number of linear models times the dimension of the contexts is more than the total budget of samples. Simulations with real data suggest that Trace-UCB is remarkably robust, outperforming a number of baselines even when its assumptions are violated.

1. Introduction

We study the problem faced by a decision-maker whose goal is to estimate a number of regression problems equally well (i.e., with a small prediction error for each of them), and has to adaptively allocate a limited budget of samples to the problems in order to gather information and improve its estimates. Two aspects of the problem formulation are key and drive the algorithm design: **1)** The observations Y collected from each regression problem depend on side information (i.e., contexts $X \in \mathbb{R}^d$) and we model the relationship between X and Y in each problem i as a linear function with unknown parameters $\beta_i \in \mathbb{R}^d$, and **2)** The "hardness" of learning each parameter β_i is unknown in advance and may vary across the problems. In particular, we assume that the observations are corrupted by noise levels that are problem-dependent and must be learned as well.

This scenario may arise in a number of different domains where a fixed experimentation budget (number of samples) should be allocated to different problems. Imagine a drug company that has developed several treatments for a particular form of disease. Now it is interested in having an accurate estimate of the performance of each of these treatments for a specific population of patients (e.g., at a particular geographical location). Given the budget allocated to this experiment, a number of patients n can participate in the clinical trial. Volunteered patients arrive sequentially over time and they are represented by a context $X \in \mathbb{R}^d$ summarizing their profile. We model the health status of patient X after being assigned to treatment i by scalar $Y_i \in \mathbb{R}$, which depends on the specific drug through a linear function with parameter β_i (i.e., $Y_i \approx X^\mathsf{T}\beta_i$). The goal is to assign each incoming patient to a treatment in such a way that at the end of the trial, we have an accurate estimate for all β_i's. This will allow us to reliably predict the expected health status of each new patient X for any treatment i. Since the parameters β_i and the noise levels are initially unknown, achieving this goal requires an adaptive allocation strategy for the n patients. Note that while n may be relatively small, as the ethical and financial costs of treating a patient are high, the distribution of the contexts X (e.g., the biomarkers of cancer patients) can be precisely estimated in advance.

This setting is clearly related to the problem of pure exploration and active learning in multi-armed bandits (Antos et al., 2008), where the learner wants to estimate the mean of a finite set of arms by allocating a finite budget of n pulls. Antos et al. (2008) first introduced this setting where the objective is to minimize the largest mean square error (MSE) in estimating the value of each arm. While the optimal solution is trivially to allocate the pulls proportionally to the variance of the arms, when the variances are unknown an exploration-exploitation dilemma arises, where variance and value of the arms must be estimated at the same time in order to allocate pulls where they are more needed (i.e., arms with high variance). Antos et al. (2008) proposed a forcing algorithm where all arms are pulled at least $\sqrt{n}$ times before allocating pulls proportionally to the estimated variances. They derived bounds on the regret, measuring the difference between the MSEs of the learn-

[1] Stanford University, Stanford, CA, USA. [2] DeepMind, Mountain View, CA, USA (The work was done when the author was with Adobe Research). [3] Inria Lille, France. Correspondence to: Carlos Riquelme <rikel@stanford.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

ing algorithm and an optimal allocation showing that the regret decreases as $O(n^{-3/2})$. A similar result is obtained by Carpentier et al. (2011) that proposed two algorithms that use upper confidence bounds on the variance to estimate the MSE of each arm and select the arm with the larger MSE at each step. When the arms are embedded in $\mathbb{R}^d$ and their mean is a linear combination with an unknown parameter, then the problem becomes an optimal experimental design problem (Pukelsheim, 2006), where the objective is to estimate the linear parameter and minimize the prediction error over all arms (see e.g., Wiens & Li 2014; Sabato & Munos 2014). In this paper, we consider an orthogonal extension to the original problem where a finite number of linear regression problems is available (i.e., the arms) and random contexts are observed at each time step. Similarly to the setting of Antos et al. (2008), we assume each problem is characterized by a noise with different variance and the objective is to return regularized least-squares (RLS) estimates with small prediction error (i.e., MSE). While we leverage on the solution proposed by Carpentier et al. (2011) to deal with the unknown variances, in our setting the presence of random contexts make the estimation problem considerably more difficult. In fact, the MSE in one specific regression problem is not only determined by the variance of the noise and the number of samples used to compute the RLS estimate, but also by the contexts observed over time.

Contributions. We propose TRACE-UCB, an algorithm that simultaneously learns the "hardness" of each problem, allocates observations proportionally to these estimates, and balances contexts across problems. We derive performance bounds for TRACE-UCB in expectation and high-probability, and compare the algorithm with several baselines. TRACE-UCB performs remarkably well in scenarios where the dimension of the contexts or the number of instances is large compared to the total budget, motivating the study of the high-dimensional setting, whose analysis and performance bounds are reported in App. F of Riquelme et al. (2017a). Finally, we provide simulations with synthetic data that support our theoretical results, and with real data that demonstrate the robustness of our approach even when some of the assumptions do not hold.

2. Preliminaries

The problem. We consider m linear regression problems, where each instance $i \in [m] = \{1, \dots, m\}$ is characterized by a parameter $\beta_i \in \mathbb{R}^d$ such that for any context $X \in \mathbb{R}^d$, a random observation $Y \in \mathbb{R}$ is obtained as

$$Y = X^\mathsf{T}\beta_i + \epsilon_i, \tag{1}$$

where the noise ϵ_i is an i.i.d. realization of a Gaussian distribution $\mathcal{N}(0, \sigma_i^2)$. We denote by $\sigma^2_{\max} = \max_i \sigma_i^2$ and by $\overline{\sigma}^2 = 1/m \sum_i \sigma_i^2$, the largest and the average variance, respectively. We define a sequential decision-making problem over n rounds, where at each round $t \in [n]$, the learning algorithm $\mathcal{A}$ receives a context X_t drawn i.i.d. from $\mathcal{N}(0, \Sigma)$, selects an instance I_t, and observes a random sample $Y_{I_t,t}$ according to (1). By the end of the experiment, a training set $\mathcal{D}_n = \{X_t, I_t, Y_{I_t,t}\}_{t\in[n]}$ has been collected and all the m linear regression problems are solved, each problem $i \in [m]$ with its own training set $\mathcal{D}_{i,n}$ (i.e., a subset of $\mathcal{D}_n$ containing samples with $I_t = i$), and estimates of the parameters $\{\hat\beta_{i,n}\}_{i\in[m]}$ are returned. For each $\hat\beta_{i,n}$, we measure its accuracy by the mean-squared error (MSE)

$$L_{i,n}(\hat\beta_{i,n}) = \mathbb{E}_X\big[(X^\mathsf{T}\beta_i - X^\mathsf{T}\hat\beta_{i,n})^2\big] = \|\beta_i - \hat\beta_{i,n}\|^2_\Sigma. \tag{2}$$

We evaluate the overall accuracy of the estimates returned by the algorithm $\mathcal{A}$ as

$$L_n(\mathcal{A}) = \max_{i\in[m]} \mathbb{E}_{\mathcal{D}_n}\big[L_{i,n}(\hat\beta_{i,n})\big], \tag{3}$$

where the expectation is w.r.t. the randomness of the contexts X_t and observations $Y_{i,t}$ used to compute $\hat\beta_{i,n}$. The objective is to design an algorithm $\mathcal{A}$ that minimizes the loss (3). This requires defining an allocation rule to select the instance I_t at each step t and the algorithm to compute the estimates $\hat\beta_{i,n}$, e.g., ordinary least-squares (OLS), regularized least-squares (RLS), or Lasso. In designing a learning algorithm, we rely on the following assumption.

Assumption 1. *The covariance matrix Σ of the Gaussian distribution generating the contexts $\{X_t\}_{t=1}^n$ is known.*

This is a standard assumption in active learning, since in this setting the learner has access to the input distribution and the main question is for which context she should ask for a label (Sabato & Munos, 2014; Riquelme et al., 2017b). Often times, companies, like the drug company considered in the introduction, own enough data to have an accurate estimate of the distribution of their customers (patients).

While in the rest of the paper we focus on $L_n(\mathcal{A})$, our algorithm and analysis can be easily extended to similar objectives such as replacing the maximum in (3) with average across all instances, i.e., $1/m \sum_{i=1}^m \mathbb{E}_{\mathcal{D}_n}\big[L_{i,n}(\hat\beta_{i,n})\big]$, and using weighted errors, i.e., $\max_i w_i\, \mathbb{E}_{\mathcal{D}_n}\big[L_{i,n}(\hat\beta_{i,n})\big]$, by updating the score to focus on the estimated standard deviation and by including the weights in the score, respectively. Later in the paper, we also consider the case where the expectation in (3) is replaced by the high-probability error (see Eq. 17).

Optimal static allocation with OLS estimates. While the distribution of the contexts is fixed and does not depend on the instance i, the errors $L_{i,n}(\hat\beta_{i,n})$ directly depend on the variances σ_i^2 of the noise ϵ_i. We define an optimal baseline obtained when the noise variances $\{\sigma_i^2\}_{i=1}^m$ are known. In particular, we focus on a static allocation algorithm $\mathcal{A}_{\text{stat}}$ that selects each instance i exactly $k_{i,n}$ times, independently of the context,[1] and returns an estimate $\hat{\beta}_{i,n}$ computed by OLS as

$$\widehat{\beta}_{i,n} = \left(\mathbf{X}_{i,n}^\mathsf{T}\mathbf{X}_{i,n}\right)^{-1}\mathbf{X}_{i,n}^\mathsf{T}\mathbf{Y}_{i,n}, \tag{4}$$

where $\mathbf{X}_{i,n} \in \mathbb{R}^{k_{i,n}\times d}$ is the matrix of (random) samples obtained at the end of the experiment, and $\mathbf{Y}_{i,n} \in \mathbb{R}^{k_{i,n}}$ is its corresponding vector of observations. It is simple to show that the global error corresponding to $\mathcal{A}_{\text{stat}}$ is

$$L_n(\mathcal{A}_{\text{stat}}) = \max_{i\in[m]} \frac{\sigma_i^2}{k_{i,n}} \mathrm{Tr}\Big(\Sigma\mathbb{E}_{\mathcal{D}_n}\big[\widehat{\Sigma}_{i,n}^{-1}\big]\Big), \tag{5}$$

where $\widehat{\Sigma}_{i,n} = \mathbf{X}_{i,n}^\mathsf{T}\mathbf{X}_{i,n}/k_{i,n} \in \mathbb{R}^{d\times d}$ is the empirical covariance matrix of the contexts assigned to instance i. Since the algorithm does not change the allocation depending on the contexts and $X_t \sim \mathcal{N}(0,\Sigma)$, $\widehat{\Sigma}_{i,n}^{-1}$ is distributed as an inverse-Wishart and we may write (5) as

$$L_n(\mathcal{A}_{\text{stat}}) = \max_{i\in[m]} \frac{d\sigma_i^2}{k_{i,n}-d-1}. \tag{6}$$

Thus, we derive the following proposition for the optimal static allocation algorithm $\mathcal{A}_{\text{stat}}^*$.

Proposition 1. *Given m linear regression problems, each characterized by a parameter β_i, Gaussian noise with variance σ_i^2, and Gaussian contexts with covariance Σ, let $n > m(d+1)$, then the optimal OLS static allocation algorithm $\mathcal{A}_{stat}^*$ selects each instance*

$$k_{i,n}^* = \frac{\sigma_i^2}{\sum_j \sigma_j^2}\, n + (d+1)\, \left(1 - \frac{\sigma_i^2}{\overline{\sigma}^2}\right), \tag{7}$$

times (up to rounding effects), and incurs the global error

$$L_n^* = L_n(\mathcal{A}_{stat}^*) = \overline{\sigma}^2\frac{md}{n} + O\left(\overline{\sigma}^2\left(\frac{md}{n}\right)^2\right). \tag{8}$$

Proof. See Appendix A.1.[2] □

Proposition 1 divides the problems into two types: those for which $\sigma_i^2 \geq \bar{\sigma}^2$ (*wild* instances) and those for which $\sigma_i^2 < \bar{\sigma}^2$ (*mild* instances). We see that for the first type, the second term in (7) is negative and the instance should be selected less frequently than in the context-free case (where the optimal allocation is given just by the first term). On the other hand, instances whose variance is below the mean variance should be pulled more often. In any case, we see that the correction to the context-free allocation (i.e., the second term) is *constant*, as it does not depend on n. Nonetheless, it does depend on d and this suggests that in high-dimensional problems, it may significantly skew the optimal allocation.

While $\mathcal{A}_{\text{stat}}^*$ effectively minimizes the prediction loss L_n, it cannot be implemented in practice since the optimal allocation k_i^* requires the variances σ_i^2 to be known at the beginning of the experiment. As a result, we need to devise a learning algorithm $\mathcal{A}$ whose performance approaches L_n^* as n increases. More formally, we define the regret of $\mathcal{A}$ as

$$R_n(\mathcal{A}) = L_n(\mathcal{A}) - L_n(\mathcal{A}_{\text{stat}}^*) = L_n(\mathcal{A}) - L_n^*, \tag{9}$$

and we expect $R_n(\mathcal{A}) = o(1/n)$. In fact, any allocation strategy that selects each instance a linear number of times (e.g., uniform sampling) achieves a loss $L_n = O(1/n)$, and thus, a regret of order $O(1/n)$. However, we expect that the loss of an effective learning algorithm decreases not just at the same rate as L_n^* but also with the very same constant, thus implying a regret that decreases faster than $O(1/n)$.

3. The TRACE-UCB Algorithm

In this section, we present and analyze an algorithm of the form discussed at the end of Section 2, which we call TRACE-UCB, whose pseudocode is in Algorithm 1.

Algorithm 1 TRACE-UCB Algorithm

```
for $i = 1, \ldots, m$ do
    Select problem instance $i$ exactly $d+1$ times
    Compute its OLS estimates $\hat{\beta}_{i,m(d+1)}$ and $\hat{\sigma}^2_{i,m(d+1)}$
end for
for steps $t = m(d+1)+1, \ldots, n$ do
    for problem instance $1 \leq i \leq m$ do
        Compute score          ($\Delta_{i,t-1}$ is defined in (11))
            $s_{i,t-1} = \frac{\widehat{\sigma}^2_{i,t-1} + \Delta_{i,t-1}}{k_{i,t-1}} \mathrm{Tr}\big(\Sigma\hat{\Sigma}^{-1}_{i,t-1}\big)$
    end for
    Select problem instance $I_t = \arg\max_{i\in[m]} s_{i,t-1}$
    Observe $X_t$ and $Y_{I_t,t}$
    Update its OLS estimators $\hat{\beta}_{I_t,t}$ and $\hat{\sigma}^2_{I_t,t}$
end for
Return RLS estimates $\{\hat{\beta}^\lambda_{i,n}\}_{i=1}^m$ with regularization $\lambda$
```

The regularization parameter $\lambda = O(1/n)$ is provided to the algorithm as input, while in practice one could set λ independently for each arm using cross-validation.

Intuition. Equation (6) suggests that while the parameters of the context distribution, particularly its covariance Σ, do

[1] This strategy can be obtained by simply selecting the first instance $k_{1,n}$ times, the second one $k_{2,n}$ times, and so on.

[2] All the proofs can be found in the appendices of the extended version of the paper (Riquelme et al., 2017a).

not impact the prediction error, the noise variances play the most important role in the loss of each problem instance. This is in fact confirmed by the optimal allocation $k_{i,n}^*$ in (7), where only the variances σ_i^2 appear. This evidence suggests that an algorithm similar to GAFS-MAX (Antos et al., 2008) or CH-AS (Carpentier et al., 2011), which were designed for the context-free case (i.e., each instance i is associated to an expected value and not a linear function) would be effective in this setting as well. Nonetheless, (6) holds only for static allocation algorithms that completely ignore the context and the history to decide which instance I_t to choose at time t. On the other hand, adaptive learning algorithms create a strong correlation between the dataset $\mathcal{D}_{t-1}$ collected so far, the current context X_t, and the decision I_t. As a result, the sample matrix $\mathbf{X}_{i,t}$ is no longer a random variable independent of $\mathcal{A}$, and using (6) to design a learning algorithm is not convenient, since the impact of the contexts on the error is completely overlooked. Unfortunately, in general, it is very difficult to study the potential correlation between the contexts $\mathbf{X}_{i,t}$, the intermediate estimates $\widehat{\beta}_{i,t}$, and the most suitable choice I_t. However, in the next lemma, we show that if at each step t, we select I_t as a function of $\mathcal{D}_{t-1}$, and *not* X_t, we may still recover an expression for the final loss that we can use as a basis for the construction of an effective learning algorithm.

Lemma 2. *Let $\mathcal{A}$ be a learning algorithm that selects the instances I_t as a function of the previous history, i.e., $\mathcal{D}_{t-1} = \{X_1, I_1, Y_{I_1,1}, \ldots, X_{t-1}, I_{t-1}, Y_{I_{t-1},t-1}\}$ and computes estimates $\widehat{\beta}_{i,n}$ using OLS. Then, its loss after n steps can be expressed as*

$$L_n(\mathcal{A}) = \max_{i\in[m]} \mathbb{E}_{\mathcal{D}_n}\left[\frac{\sigma_i^2}{k_{i,n}} \mathrm{Tr}\Big(\Sigma\widehat{\Sigma}_{i,n}^{-1}\Big)\right], \qquad (10)$$

where $k_{i,n} = \sum_{t=1}^n \mathbb{I}\{I_t = i\}$ and $\widehat{\Sigma}_{i,n} = \mathbf{X}_{i,n}^\mathsf{T}\mathbf{X}_{i,n}/k_{i,n}$.

Proof. See Appendix B. □

Remark 1 (assumptions). We assume noise and contexts are Gaussian. The noise Gaussianity is crucial for the estimates of the parameter $\widehat{\beta}_{i,t}$ and variance $\widehat{\sigma}_{i,t}^2$ to be independent of each other, for each instance i and time t (we actually need and derive a stronger result in Lemma 9, see Appendix B). This is key in proving Lemma 2, as it allows us to derive a closed form expression for the loss function which holds under our algorithm, and is written in terms of the number of pulls and the trace of the inverse empirical covariance matrix. Note that $\widehat{\beta}_{i,t}$ drives our loss, while $\widehat{\sigma}_{i,t}^2$ drives our decisions. One way to remove this assumption is by defining and directly optimizing a surrogate loss equal to (10) instead of (3). On the other hand, the Gaussianity of contexts leads to the whitened inverse covariance estimate $\Sigma\widehat{\Sigma}_{i,n}^{-1}$ being distributed as an inverse Wishart. As there is a convenient closed formula for its mean, we can find the exact optimal static allocation $k_{i,n}^*$ in Proposition 1, see (7). In general, for sub-Gaussian contexts, no such closed formula for the trace is available. However, as long as the optimal allocation $k_{i,n}^*$ has no second order n^α terms for $1/2 \leq \alpha < 1$, it is possible to derive the same regret rate results that we prove later on for TRACE-UCB.

Equation (10) makes it explicit that the prediction error comes from two different sources. The first one is the noise in the measurements $\mathbf{Y}$, whose impact is controlled by the unknown variances σ_i^2's. Clearly, the larger the σ_i^2 is, the more observations are required to achieve the desired accuracy. At the same time, the *diversity* of contexts across instances also impacts the overall prediction error. This is very intuitive, since it would be a terrible idea for the research center discussed in the introduction to estimate the parameters of a drug by providing the treatment only to a hundred almost identical patients. We say contexts are balanced when $\widehat{\Sigma}_{i,n}$ is well conditioned. Therefore, a good algorithm should take care of both aspects.

There are two extreme scenarios regarding the contributions of the two sources of error. **1)** If the number of contexts n is relatively large, since the context distribution is fixed, one can expect that contexts allocated to each instance eventually become balanced (i.e., TRACE-UCB does not bias the distribution of the contexts). In this case, it is the difference in σ_i^2's that drives the number of times each instance is selected. **2)** When the dimension d or the number of arms m is large w.r.t. n, balancing contexts becomes critical, and can play an important role in the final prediction error, whereas the σ_i^2's are less relevant in this scenario. While a learning algorithm cannot deliberately choose a specific context (i.e., X_t is a random variable), we may need to favor instances in which the contexts are poorly balanced and their prediction error is large, despite the fact that they might have small noise variances.

Algorithm. TRACE-UCB is designed as a combination of the upper-confidence-bound strategy used in CH-AS (Carpentier et al., 2011) and the loss in (10), so as to obtain a learning algorithm capable of allocating according to the estimated variances and at the same time balancing the error generated by context mismatch. We recall that all the quantities that are computed at every step of the algorithm are indexed at the beginning and end of a step t by $i, t-1$ (e.g., $\widehat{\sigma}_{i,t-1}^2$) and i, t (e.g., $\widehat{\beta}_{i,t}$), respectively. At the end of each step t, TRACE-UCB first computes an OLS estimate $\widehat{\beta}_{i,t}$, and then use it to estimate the variance $\widehat{\sigma}_{i,t}^2$ as

$$\widehat{\sigma}_{i,t}^2 = \frac{1}{k_{i,t} - d}\big\|\mathbf{Y}_{i,t} - \mathbf{X}_{i,t}^\mathsf{T}\widehat{\beta}_{i,t}\big\|^2,$$

which is the average squared deviation of the predictions based on $\widehat{\beta}_{i,t}$. We rely on the following concentration inequality for the variance estimate of linear regression with Gaussian noise, whose proof is reported in Appendix C.1.

Proposition 3. *Let the number of pulls $k_{i,t} \geq d+1$ and $R \geq \max_i \sigma_i^2$. If $\delta \in (0, 3/4)$, then for any instance i and step $t > m(d+1)$, with probability at least $1 - \frac{\delta}{2}$, we have*

$$|\hat{\sigma}_{i,t}^2 - \sigma_i^2| \leq \Delta_{i,t} \triangleq R\sqrt{\frac{64}{k_{i,t}-d}\left(\log\frac{2mn}{\delta}\right)^2}. \tag{11}$$

Given (11), we can construct an upper-bound on the prediction error of any instance i and time step t as

$$s_{i,t-1} = \frac{\hat{\sigma}_{i,t-1}^2 + \Delta_{i,t-1}}{k_{i,t-1}} \operatorname{Tr}\left(\Sigma\widehat{\Sigma}_{i,t-1}^{-1}\right), \tag{12}$$

and then simply select the instance which maximizes this score, i.e., $I_t = \arg\max_i s_{i,t-1}$. Intuitively, TRACE-UCB favors problems where the prediction error is potentially large, either because of a large noise variance or because of significant unbalance in the observed contexts w.r.t. the target distribution with covariance Σ. A subtle but critical aspect of TRACE-UCB is that by ignoring the current context X_t (but using all the past samples $\mathbf{X}_{t-1}$) when choosing I_t, the distribution of the contexts allocated to each instance stays untouched and the second term in the score $s_{i,t-1}$, i.e., $\operatorname{Tr}(\Sigma\widehat{\Sigma}_{i,t-1}^{-1})$, naturally tends to d as more and more (random) contexts are allocated to instance i. This is shown by Proposition 4 whose proof is in Appendix C.2.

Proposition 4. *Force the number of samples $k_{i,t} \geq d+1$. If $\delta \in (0,1)$, for any $i \in [m]$ and step $t > m(d+1)$ with probability at least $1 - \delta/2$, we have*

$$\left(1 - C_{\mathrm{Tr}}\sqrt{\frac{d}{k_{i,t}}}\right)^2 \leq \frac{\operatorname{Tr}\left(\Sigma\widehat{\Sigma}_{i,t}^{-1}\right)}{d} \leq \left(1 + 2C_{\mathrm{Tr}}\sqrt{\frac{d}{k_{i,t}}}\right)^2,$$

with $C_{\mathrm{Tr}} = 1 + \sqrt{2\log(4nm/\delta)/d}$.

While Proposition 4 shows that the error term due to context mismatch tends to the constant d for all instances i as the number of samples tends to infinity, when t is small w.r.t. d and m, correcting for the context mismatch may significantly improve the accuracy of the estimates $\widehat{\beta}_{i,n}$ returned by the algorithm. Finally, note that while TRACE-UCB uses OLS to compute estimates $\widehat{\beta}_{i,t}$, it computes its returned parameters $\widehat{\beta}_{i,n}$ by ridge regression (RLS) with regularization parameter λ as

$$\hat{\beta}_i^{\lambda} = (\mathbf{X}_{i,n}^{\mathsf{T}}\mathbf{X}_{i,n} + \lambda\mathbf{I})^{-1}\mathbf{X}_{i,n}^{\mathsf{T}}\mathbf{Y}_{i,n}. \tag{13}$$

As we will discuss later, using RLS makes the algorithm more robust and is crucial in obtaining regret bounds both in expectation and high probability.

Performance Analysis. Before proving a regret bound for TRACE-UCB, we report an intermediate result (proof in App. D.1) that shows that TRACE-UCB *behaves* similarly to the optimal static allocation.

Theorem 5. *Let $\delta > 0$. With probability at least $1 - \delta$, the total number of contexts that TRACE-UCB allocates to each problem instance i after n rounds satisfies*

$$k_{i,n} \geq k_{i,n}^* - \frac{C_\Delta + 8C_{\mathrm{Tr}}}{\sigma_{\min}^2}\sqrt{\frac{nd}{\lambda_{\min}}} - \Omega(n^{1/4}) \tag{14}$$

where $R \geq \sigma_{\max}^2$ is known by the algorithm, and we defined $C_\Delta = 16R\log(2mn/\delta)$ and $\lambda_{\min} = \sigma_{\min}^2 / \sum_j \sigma_j^2$.

We now report our regret bound for the TRACE-UCB algorithm. The proof of Theorem 6 is in Appendix D.2.

Theorem 6. *The regret of the Trace-UCB algorithm, i.e., the difference between its loss and the loss of optimal static allocation (see Eq.* (8)*), is upper-bounded by*

$$L_n(\mathcal{A}) - L_n^* \leq O\left(\frac{1}{\sigma_{\min}^2}\left(\frac{d}{\lambda_{\min}n}\right)^{3/2}\right). \tag{15}$$

Eq. (15) shows that the regret decreases as $O(n^{-3/2})$ as expected. This is consistent with the context-free results (Antos et al., 2008; Carpentier et al., 2011), where the regret decreases as $n^{-3/2}$, which is conjectured to be optimal. However, it is important to note that in the contextual case, the numerator also includes the dimensionality d. Thus, when $n \gg d$, the regret will be small, and it will be larger when $n \approx d$. This motivates studying the high-dimensional setting (App. F). Eq. (15) also indicates that the regret depends on a problem-dependent constant $1/\lambda_{\min}$, which measures the complexity of the problem. Note that when $\sigma_{\max}^2 \approx \sigma_{\min}^2$, we have $1/\lambda_{\min} \approx m$, but $1/\lambda_{\min}$ could be much larger when $\sigma_{\max}^2 \gg \sigma_{\min}^2$.

Remark 2. We introduce a baseline motivated by the context-free problem. At round t, let VAR-UCB selects the instance that maximizes the score[3]

$$s'_{i,t-1} = \frac{\hat{\sigma}_{i,t-1}^2 + \Delta_{i,t-1}}{k_{i,t-1}}. \tag{16}$$

The only difference with the score used by TRACE-UCB is the lack of the trace term in (12). Moreover, the regret of this algorithm has similar *rate* in terms of n and d as that of TRACE-UCB reported in Theorem 6. However, the simulations of Sect. 4 show that the regret of VAR-UCB is actually much higher than that of TRACE-UCB, specially when dm is close to n. Intuitively, when n is close to dm, balancing contexts becomes critical, and VAR-UCB suffers because its score does not explicitly take them into account.

Sketch of the proof of Theorem 6. The proof is divided into three parts. **1)** We show that the behavior of the ridge

[3] Note that VAR-UCB is similar to both the CH-AS and B-AS algorithms in Carpentier et al. (2011).

loss of TRACE-UCB is similar to that reported in Lemma 2 for algorithms that rely on OLS; see Lemma 19 in Appendix E. The independence of the $\hat{\beta}_{i,t}$ and $\hat{\sigma}^2_{i,t}$ estimates is again essential (see Remark 1). Although the loss of TRACE-UCB depends on the ridge estimate of the parameters $\hat{\beta}^{\lambda}_{i,n}$, the decisions made by the algorithm at each round only depend on the variance estimates $\hat{\sigma}^2_{i,t}$ and observed contexts. **2)** We follow the ideas in Carpentier et al. (2011) to lower-bound the total number of pulls $k_{i,n}$ for each $i \in [m]$ under a good event (see Theorem 5 and its proof in Appendix D.1). **3)** We finally use the ridge regularization to bound the impact of those cases *outside* the good event, and combine everything in Appendix D.2.

The regret bound of Theorem 6 shows that the largest *expected* loss across the problem instances incurred by TRACE-UCB quickly approaches the loss of the optimal static allocation algorithm (which knows the true noise variances). While $L_n(\mathcal{A})$ measures the worst *expected* loss, at any specific *realization* of the algorithm, there may be one of the instances which is very poorly estimated. As a result, it would also be desirable to obtain guarantees for the (random) maximum loss

$$\widetilde{L}_n(\mathcal{A}) = \max_{i\in[m]} \|\beta_i - \hat{\beta}_{i,n}\|^2_{\Sigma}. \tag{17}$$

In particular, we are able to prove the following high-probability bound on $\widetilde{L}_n(\mathcal{A})$ for TRACE-UCB.

Theorem 7. *Let $\delta > 0$, and assume $\|\beta_i\|_2 \leq Z$ for all i, for some $Z > 0$. With probability at least $1 - \delta$,*

$$\widetilde{L}_n \leq \frac{\sum_{j=1}^{m} \sigma_j^2}{n}\Big(d + 2\log\frac{3m}{\delta}\Big) + O\bigg(\frac{1}{\sigma^2_{\min}}\Big(\frac{d}{n\lambda_{\min}}\Big)^{\frac{3}{2}}\bigg). \tag{18}$$

Note that the first term in (18) corresponds to the first term of the loss for the optimal static allocation, and the second term is, again, a $n^{-3/2}$ deviation. However, in this case, the guarantees hold *simultaneously* for all the instances.

Sketch of the proof of Theorem 7. In the proof we slightly modify the confidence ellipsoids for the $\hat{\beta}_{i,t}$'s, based on self-normalized martingales, and derived in (Abbasi-Yadkori et al., 2011); see Thm. 13 in App. C. By means of the confidence ellipsoids we control the loss in (17). Their radiuses depend on the number of samples per instance, and we rely on a high-probability events to compute a lower bound on the number of samples. In addition, we need to make sure the mean norm of the contexts will not be too large (see Corollary 15 in App. C). Finally, we combine the lower bound on $k_{i,n}$ with the confidence ellipsoids to conclude the desired high-probability guarantees in Thm. 7.

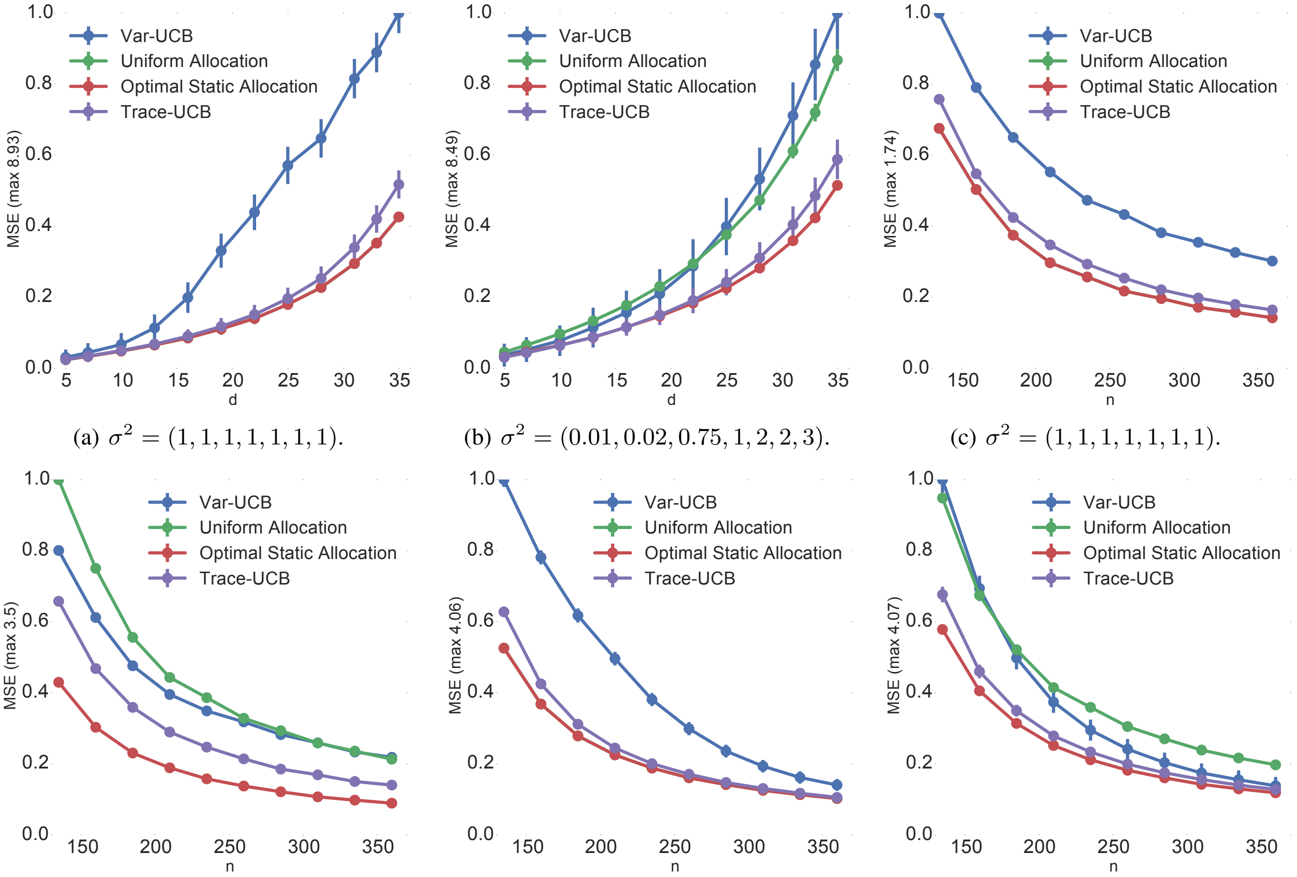
High-Dimensional Setting. High-dimensional linear models are quite common in practice, motivating the study of the $n < dm$ case, where the algorithms discussed so far break down. We propose SPARSE-TRACE-UCB in Appendix F, an extension of TRACE-UCB that assumes and takes advantage of *joint* sparsity across the linear functions. The algorithm has two-stages: first, an approximate support is recovered, and then, TRACE-UCB is applied to the induced lower dimensional space. We discuss and extend our high-probability guarantees to SPARSE-TRACE-UCB under suitable standard assumptions in Appendix F.

4. Simulations

In this section, we provide empirical evidence to support our theoretical results. We consider both synthetic and real-world problems, and compare the performance (in terms of normalized MSE) of TRACE-UCB to uniform sampling, optimal static allocation (which requires the knowledge of noise variances), and the context-free algorithm VAR-UCB (see Remark 2). We do not compare to GFSP-MAX and GAFS-MAX (Antos et al., 2008) since they are outperformed by CH-AS Carpentier et al. (2011) and VAR-UCB is the same as CH-AS, except for the fact that we use the concentration inequality in Prop. 3, since we are estimating the variance from a regression problem using OLS.

First, we use synthetic data to ensure that all the assumptions of our model are satisfied, namely we deal with linear regression models with Gaussian context and noise. We set the number of problem instances to $m = 7$ and consider two scenarios: one in which all the noise variances are equal to 1 and one where they are *not* equal, and $\sigma^2 = (0.01, 0.02, 0.75, 1, 2, 2, 3)$. In the latter case, $\sigma^2_{\max}/\sigma^2_{\min} = 300$. We study the impact of (independently) increasing dimension d and horizon n on the performance, while keeping all other parameters fixed. Second, we consider real-world datasets in which the underlying model is non-linear and the contexts are not Gaussian, to observe how TRACE-UCB behaves (relative to the baselines) in settings where its main underlying assumptions are violated.

Synthetic Data. In Figures 1(a,b), we display the results for fixed horizon $n = 350$ and increasing dimension d. For each value of d, we run $10,000$ simulations and report the median of the maximum error across the instances for each simulation. In Fig. 1(a), where σ_i^2's are equal, uniform sampling and optimal static allocation execute the same allocation since there is no difference in the expected losses of different instances. Nonetheless we notice that VAR-UCB suffers from poor estimation as soon as d increases, while TRACE-UCB is competitive with the optimal performance. This difference in performance can be explained by the fact that VAR-UCB does not control for contextual balance, which becomes a dominant factor in the loss of a learning strategy for problems of high dimensionality. In Fig. 1(b), in which σ_i^2's are different, uniform sampling is no longer optimal but even in this case VAR-UCB performs



(a) $\sigma^2 = (1, 1, 1, 1, 1, 1, 1)$.

(b) $\sigma^2 = (0.01, 0.02, 0.75, 1, 2, 2, 3)$.

(c) $\sigma^2 = (1, 1, 1, 1, 1, 1, 1)$.

(d) $\sigma^2 = (0.01, 0.02, 0.75, 1, 2, 2, 3)$.

(e) $\sigma^2 = (1, 1, 1, 1, 1, 1, 1)$.

(f) $\sigma^2 = (0.01, 0.02, 0.75, 1, 2, 2, 3)$.

Figure 1. White Gaussian synthetic data with $m = 7$. In Figures (a,b), we set $n = 350$. In Figures (c,d,e,f), we set $d = 10$.

better than uniform sampling only for small $d < 23$, where it is more important to control for the σ_i^2's. For larger dimensions, balancing uniformly the contexts eventually becomes a better strategy, and uniform sampling outperforms VAR-UCB. In this case too, TRACE-UCB is competitive with the optimal static allocation even for large d, successfully balancing both noise variance and contextual error.

Next, we study the performance of the algorithms w.r.t. n. We report two different losses, one in expectation (3) and one in high probability (17), corresponding to the results we proved in Theorems 6 and 7, respectively. In order to approximate the loss in (3) (Figures 1(c,d)) we run $30,000$ simulations, compute the average prediction error for each instance $i \in [m]$, and finally report the maximum mean error across the instances. On the other hand, we estimate the loss in (17) (Figures 1(e,f)) by running $30,000$ simulations, taking the maximum prediction error across the instances for each simulation, and finally reporting their median.

In Figures 1(c, d), we display the loss for fixed dimension $d = 10$ and horizon from $n = 115$ to 360. In Figure 1(c), TRACE-UCB performs similarly to the optimal static allocation, whereas VAR-UCB performs significantly worse, ranging from 25% to 50% higher errors than TRACE-UCB, due to some catastrophic errors arising from unlucky contextual realizations for an instance. In Fig. 1(d), as the number of contexts grows, uniform sampling's simple context balancing approach is enough to perform as well as VAR-UCB that again heavily suffers from large mistakes. In both figures, TRACE-UCB smoothly learns the σ_i^2's and outperforms uniform sampling and VAR-UCB. Its performance is comparable to that of the optimal static allocation, especially in the case of equal variances in Fig. 1(c).

In Figure 1(e), TRACE-UCB learns and properly balances observations extremely fast and obtains an almost optimal performance. Similarly to figures 1(a,c), VAR-UCB struggles when variances $\hat{\sigma}_i^2$ are almost equal, mainly because it gets confused by random deviations in variance estimates $\hat{\sigma}_i^2$, while overlooking potential and harmful context imbalances. Note that even when $n = 360$ (rightmost point), its median error is still 25% higher than TRACE-UCB's. In Fig. 1(f), as expected, uniform sampling performs poorly, due to mismatch in variances, and only outperforms VAR-UCB for small horizons in which uniform allocation pays off. On the other hand, TRACE-UCB is able to success-

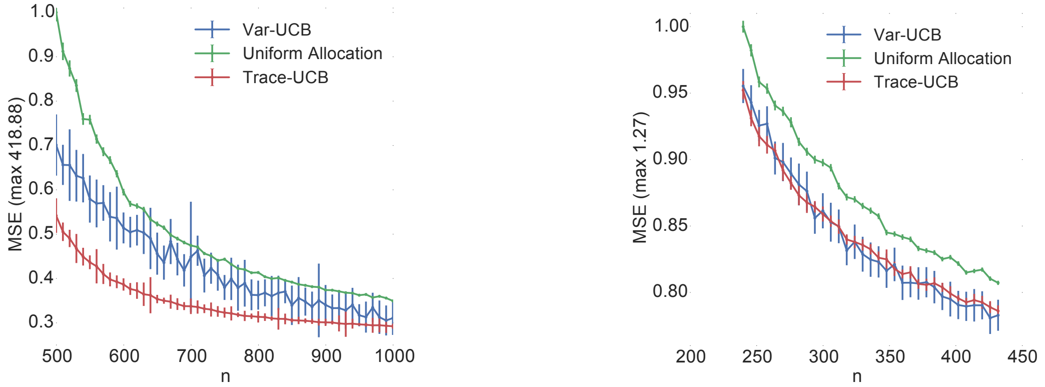


Figure 2. Results on Jester *(left)* with $d\!=\!40, m\!=\!10$ and MovieLens *(right)* with $d\!=\!25, m\!=\!5$. Median over 1000 simulations.

fully handle the tradeoff between learning and allocating according to variance estimates $\hat{\sigma}_i^2$, while accounting for the contextual trace $\widehat{\Sigma}_i$, even for very low n. We observe that for large n, VAR-UCB eventually reaches the performance of the optimal static allocation and TRACE-UCB.

In practice the loss in (17) (figures 1(e,f)) is often more relevant than (3), since it is in high probability and not in expectation, and TRACE-UCB shows excellent performance and robustness, regardless of the underlying variances σ_i^2.

Real Data. TRACE-UCB is based on assumptions such as linearity, and Gaussianity of noise and context that may not hold in practice, where data may show complex dependencies. Therefore, it is important to evaluate the algorithm with real-world data to see its robustness to the violation of its assumptions. We consider two collaborative filtering datasets in which users provide ratings for items. We choose a dense subset of k users and p items, where every user has rated every item. Thus, each user is represented by a p-dimensional vector of ratings. We define the user context by d out of her p ratings, and learn to predict her remaining $m = p - d$ ratings (each one is a problem instance). All item ratings are first centered, so each item's mean is zero. In each simulation, n out of the k users are selected at random to be fed to the algorithm, also in random order. Algorithms can select any instance as the dataset contains the ratings of every instance for all the users. At the end of each simulation, we compute the prediction error for each instance by using the $k - n$ users that did not participate in training for that simulation. Finally, we report the median error across all simulations.

Fig. 2(a) reports the results using the Jester Dataset by (Goldberg et al., 2001) that consists of joke ratings in a continuous scale from -10 to 10. We take $d = 40$ joke ratings as context and learn the ratings for another 9 jokes. In addition, we add another function that counts the total number of movies originally rated by the user. The latter is also centered, bounded to the same scale, and has higher variance (without conditioning on X). The number of total users is $k = 3811$, and $m = 10$. When the number of observations is limited, the advantage of TRACE-UCB is quite significant (the improvement w.r.t. uniform allocation goes from 45% to almost 20% for large n, while w.r.t. VAR-UCB it goes from almost 30% to roughly 5%), even though the model and context distribution are far from linear and Gaussian, respectively.

Fig. 2(b) shows the results for the MovieLens dataset (Maxwell Harper & Konstan, 2016) that consists of movie ratings between 0 and 5 with 0.5 increments. We select 30 popular movies rated by $k = 1363$ users, and randomly choose $m = 5$ of them to learn (so $d = 25$). In this case, all problems have similar variance ($\hat{\sigma}^2_{\max}/\hat{\sigma}^2_{\min} \approx 1.3$) so uniform allocation seems appropriate. Both TRACE-UCB and VAR-UCB modestly improve uniform allocation, while their performance is similar.

5. Conclusions

We studied the problem of adaptive allocation of n contextual samples of dimension d to estimate m linear functions equally well, under heterogenous noise levels σ_i^2 that depend on the linear instance and are unknown to the decision-maker. We proposed TRACE-UCB, an optimistic algorithm that successfully solves the exploration-exploitation dilemma by simultaneously learning the σ_i^2's, allocating samples accordingly to their estimates, and balancing the contextual information across the instances. We also provide strong theoretical guarantees for two losses of interest: in expectation and high-probability. Simulations were conducted in several settings, with both synthetic and real data. The favorable results suggest that TRACE-UCB is reliable, and remarkably robust even in settings that fall outside its assumptions, thus, a useful and simple tool to implement in practice.

Acknowledgements. A. Lazaric is supported by French Ministry of Higher Education and Research, Nord-Pas-de-Calais Regional Council and French National Research Agency projects ExTra-Learn (n.ANR-14-CE24-0010-01).

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, Cs. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pp. 2312–2320, 2011.

Antos, A., Grover, V., and Szepesvári, Cs. Active learning in multi-armed bandits. In *International Conference on Algorithmic Learning Theory*, pp. 287–302, 2008.

Carpentier, A., Lazaric, A., Ghavamzadeh, M., Munos, R., and Auer, P. Upper-confidence-bound algorithms for active learning in multi-armed bandits. In *Algorithmic Learning Theory*, pp. 189–203. Springer, 2011.

Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

Hastie, T., Tibshirani, R., and Wainwright, M. *Statistical learning with sparsity: the lasso and generalizations*. CRC Press, 2015.

Maxwell Harper, F. and Konstan, J. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2016.

Negahban, S. and Wainwright, M. Simultaneous support recovery in high dimensions: Benefits and perils of block-regularization. *IEEE Transactions on Information Theory*, 57(6):3841–3863, 2011.

Obozinski, G., Wainwright, M., and Jordan, M. Support union recovery in high-dimensional multivariate regression. *The Annals of Statistics*, pp. 1–47, 2011.

Pukelsheim, F. *Optimal Design of Experiments*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2006.

Raskutti, G., Wainwright, M. J, and Yu, B. Restricted eigenvalue properties for correlated gaussian designs. *Journal of Machine Learning Research*, 11(8):2241–2259, 2010.

Riquelme, C., Ghavamzadeh, M., and Lazaric, A. Active learning for accurate estimation of linear models. *arXiv preprint arXiv:1703.00579*, 2017a.

Riquelme, C., Johari, R., and Zhang, B. Online active linear regression via thresholding. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017b.

Sabato, S. and Munos, R. Active regression by stratification. In *Advances in Neural Information Processing Systems*, pp. 469–477, 2014.

Vershynin, R. Introduction to the non-asymptotic analysis of random matrices. *arXiv:1011.3027*, 2010.

Wainwright, M. *High-dimensional statistics: A non-asymptotic viewpoint*. Draft, 2015.

Wang, W., Liang, Y., and Xing, E. Block regularized lasso for multivariate multi-response linear regression. In *AISTATS*, 2013.

Wiens, D. and Li, P. V-optimal designs for heteroscedastic regression. *Journal of Statistical Planning and Inference*, 145:125–138, 2014.