Quantum Algorithms for the Pathwise Lasso

Joao F. Doriguello[1,2], Debbie Lim[2,3], Chi Seng Pun[4], Patrick Rebentrost[2,5], and Tushar Vaidya[4]

[1]HUN-REN Alfréd Rényi Institute of Mathematics, Budapest, Hungary
[2]Centre for Quantum Technologies, National University of Singapore, Singapore
[3]Center for Quantum Computer Science, Faculty of Computing, University of Latvia, Latvia
[4]School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
[5]Department of Computer Science, National University of Singapore, Singapore

We present a novel quantum high-dimensional linear regression algorithm with an ℓ_1-penalty based on the classical LARS (Least Angle Regression) pathwise algorithm. Similarly to available classical numerical algorithms for Lasso, our quantum algorithm provides the full regularisation path as the penalty term varies, but quadratically faster per iteration under specific conditions. A quadratic speedup on the number of features/predictors d is possible by using the simple quantum minimum-finding subroutine from Dürr and Høyer (arXiv'96) in order to obtain the joining time at each iteration. We then improve upon this simple quantum algorithm and obtain a quadratic speedup both in the number of features d and the number of observations n by using the approximate quantum minimum-finding subroutine from Chen and de Wolf (ICALP'23). In order to do so, we construct a quantum unitary based on quantum amplitude estimation to approximately compute the joining times to be searched over by the approximate quantum minimum-finding subroutine. Since the joining times are no longer exactly computed, it is no longer clear that the resulting approximate quantum algorithm obtains a good solution. As another main contribution, we prove, via an approximate version of the KKT conditions and a duality gap, that the LARS algorithm (and therefore our quantum algorithm) is robust to errors. This means that it still outputs a path that minimises the Lasso cost function up to a small error if the joining times are only approximately computed. Furthermore, we show that, when the observations are sampled from a Gaussian distribution, our quantum algorithm's complexity only depends polylogarithmically on n, exponentially better than the classical LARS algorithm, while keeping the quadratic improvement on d. Moreover, we propose a dequantised version of our quantum algorithm that also retains the polylogarithmic dependence on n, albeit presenting the linear scaling on d from the standard LARS algorithm. Finally, we prove query lower bounds for classical and quantum Lasso algorithms.

Joao F. Doriguello: doriguello@renyi.hu, www.joaodoriguello.com
Debbie Lim: limhueychih@gmail.com
Chi Seng Pun: cspun@ntu.edu.sg
Patrick Rebentrost: patrick@comp.nus.edu.sg
Tushar Vaidya: tushar.vaidya@ntu.edu.sg

Contents

1 Introduction

One of the most important research topics that has attracted renewed attention by statisticians and the machine learning community is high-dimensional data analysis [BEM13, BvdG11, GG08, JT09, KKMR22, WM22, ZY06]. For linear regression, this means that the number of data points is less than the number of explanatory variables (features). In machine learning parlance, this is usually referred to as overparameterisation. Solutions to such problems are, however, ill defined. Having more variables to choose from is a double-edged sword. While it gives us freedom of choice, computational cost considerations favour choosing sparse models. Some sort of regularisation in the linear model, usually in the form of a penalty term, is needed to favour sparse models.

Let us consider the regression problem defined as follows. Assume to be given a fixed design matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and a vector of observations $\mathbf{y} \in \mathbb{R}^n$. The setting is such that $d \geqslant n$, i.e., more features than observations are given. Typical examples are $(n, d) = (500, 12288), (7500, 49152), (18737, 65536)$ for image reconstructions [CWB08, BJPD17, ADL+20, SE20]. In portfolio optimization, an instance would be $(n, d) = (504, 600), (252, 486)$ [CPW17, PW19]. The un-regularized ℓ_2-regression problem is defined as $\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$. Generally, different norms could be used in the penalty term. Essentially, the three popular sparse regression models reduce to three prototypes recalled below with ℓ_0, ℓ_1, ℓ_2-norms and tuning parameter λ:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_0 \quad \text{for all } \lambda > 0 \quad \text{(best subset selection)},$$

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \quad \text{for all } \lambda > 0 \quad \text{(Lasso regression)},$$

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2 \quad \text{for all } \lambda > 0 \quad \text{(Ridge regression)}.$$

The easiest is Ridge regression which has an analytical solution and the problem is convex, but tends to give non-sparse solutions. The best subset selection is not convex and is NP-hard [DK08] in general. Lasso regression with ℓ_1-penalty, on the other hand, is a convex relaxation of the best subset selection problem and will be our central object of study throughout this paper. Lasso stands for Least Absolute Shrinkage and Selection operator and has been studied extensively in the literature because of its convexity and parsimonious solutions, leading to its use across disciplines and in many different fields where sparsity is a desired prerequisite: compressed sensing [CDS01, CRT06, FNW07], genetics [UGH09, WCH+09], and finance [TYG15, CS20, PL22]. For large language models that use deep learning, overparameterisation is actually desired. In high-dimensional linear regression models, in contrast, parsimonious models are desired. The conventional statistical paradigm is to keep the number of features fixed and study the asymptotics as the sample size increases. The other paradigm is predictor selection given a large choice that fits into high-dimensional regression. Selecting which of the many variables available is part of model choice and techniques that help us in choosing the right model have significant value. Having a larger number of variables in statistical models makes it simpler to pick the right model and achieve improved predictive performance. Overparametrisation is a *virtue* and increases the likelihood of including the essential features in the model. Yet this should be balanced with regularisation [EHJT04, FL10, LTTT14]. Introducing an ℓ_1-penalty confers a number of advantages [CDS01, HTW15, Tib18]:

1. The ℓ_1-penalty provides models that can be interpreted in a simple way and thus have advantages compared to black boxes (deep neural networks) in terms of transparency and intelligibility [MSK+19, RCC+22];

2. If the original model generating the data is sparse, then an ℓ_1-penalty provides the right framework to recover the original signal;

3. Lasso selects the true model consistently even with noisy data [ZY06], and this remains an active area of research [MOK23];

4. As ℓ_1-penalties are convex, sparsity leads to computational advantages. For example, if we have two million features but only 200 observations, estimating two million parameters becomes extremely difficult.

Generally, the optimal tuning parameter λ has to be determined from the data and as such we would like an endogenous model that determines the optimal $\boldsymbol{\beta}$ along with a specific λ. The key point is that we require a path of solutions as λ varies [EHJT04, FHT10a, ZLZ18].

On the other hand, quantum computing is a new paradigm that offers faster computation for certain problems, for example, integer factorisation and discrete logarithm [Sho94, Sho99]. Quantum algorithms have been proposed for unregularised linear regression [WBL12, Wan17, CGJ19, KMTY21], Ridge regression [YGW21, SX20, Sha23, CdW23], and simple versions of Lasso regression [CdW23]. In this paper, we propose new quantum algorithms (and also dequantised versions) for the *pathwise* Lasso regression based on the famous Least Angle Regression (LARS) algorithm.

1.1 Least Angle Regression algorithm

The pathwise Lasso regression problem is the problem when the regression parameter λ varies. To be more precise, consider the Lasso estimator described above,

$$\hat{\boldsymbol{\beta}} \in \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \mathcal{L}^{(\lambda)}(\boldsymbol{\beta}) \quad \text{where} \quad \mathcal{L}^{(\lambda)}(\boldsymbol{\beta}) \triangleq \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \quad \text{for all } \lambda > 0. \tag{1}$$

The ℓ_1-penalty term is $\|\boldsymbol{\beta}\|_1 = \sum_{i=1}^d |\beta_i|$, whereas the predictive loss is the standard Euclidean squared norm. In this work, we are interested in the Lasso solution $\hat{\boldsymbol{\beta}}(\lambda)$ as a function of the regularisation parameter $\lambda > 0$. For such we define the optimal regularisation path $\mathcal{P}$:

$$\mathcal{P} \triangleq \{\hat{\boldsymbol{\beta}}(\lambda) : \lambda > 0\}.$$

Efron *et al.* [EHJT04] showed that the optimal regularisation path $\mathcal{P}$ is *piecewise linear* and *continuous* with respect to λ. This means that there exist an $m \in \mathbb{N}$ and $\infty > \lambda_0 > \cdots > \lambda_{m-1} > \lambda_m = 0$ and $\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_m \in \mathbb{R}^d$ such that[1]

$$\hat{\boldsymbol{\beta}}(\lambda) = \hat{\boldsymbol{\beta}}(\lambda_t) + (\lambda_t - \lambda)\boldsymbol{\theta}_t \quad \text{for } \lambda_{t+1} < \lambda \leqslant \lambda_t \quad (t \in \{0, \ldots, m-1\}). \tag{2}$$

There is a maximal value $\lambda_{\max} = \lambda_0$ where $\hat{\boldsymbol{\beta}}(\lambda) = \mathbf{0}$ for all $\lambda \geqslant \lambda_0$ and $\hat{\beta}_j(\lambda) \neq 0$ for some $j \in [d]$ and $\lambda < \lambda_0$. Such value is $\lambda_0 = \|\mathbf{X}^\top \mathbf{y}\|_\infty$. The $m+1$ points $\lambda_0, \ldots, \lambda_m$ where $\partial\hat{\boldsymbol{\beta}}(\lambda)/\partial\lambda$ changes are called *kinks*, and the path $\{\hat{\boldsymbol{\beta}}(\lambda) : \lambda_{t+1} < \lambda \leqslant \lambda_t\}$ between two consecutive kinks λ_{t+1} and λ_t defines a linear segment. The fact that the regularisation path is piecewise linear and the locations of the kinks can be derived from the Karush-Kuhn-Tucker (KKT) optimality conditions (see Section 3 for more details).

Since the regularisation path is piecewise linear, one needs only to compute all kinks $(\lambda_t, \hat{\boldsymbol{\beta}}(\lambda_t))$ for $t \in \{0, \ldots, m-1\}$, from which the whole regularisation path follows by linear interpolation. That is exactly what the Least Angle Regression (LARS) algorithm proposed and named by Efron *et al.* [EHJT04] does (a very similar idea appeared earlier in the works of Osborne *et al.* [OPT00b, OPT00a]). Starting at $\lambda = \infty$ (or λ_0) where the Lasso solution is $\hat{\boldsymbol{\beta}}(\lambda) = \mathbf{0} \in \mathbb{R}^d$, the LARS algorithm decreases the regularisation parameter λ and computes the regularisation path by finding the kinks along the way with the aid of the KKT conditions. Each kink corresponds to an iteration of the algorithm. More specifically, at each iteration t, the LARS algorithm computes the next kink λ_{t+1} by finding the closest point (to the previous kink) where the KKT conditions break and thus need to be updated. This is done by performing a search over the *active set* $\mathcal{A} \triangleq \{i \in [d] : \hat{\beta}_i \neq 0\}$ and another search over the *inactive set* $\mathcal{I} \triangleq [d] \backslash \mathcal{A}$. The search over $\mathcal{A}$ finds the next point $\lambda_{t+1}^{\text{cross}}$, called *crossing time*, where a variable i_{t+1}^{cross} must leave $\mathcal{A}$ and join $\mathcal{I}$. The search over $\mathcal{I}$ finds the next point $\lambda_{t+1}^{\text{join}}$, called *joining time*, where a variable i_{t+1}^{join} must leave $\mathcal{I}$ and join $\mathcal{A}$. The next kink is thus $\lambda_{t+1} = \max\{\lambda_{t+1}^{\text{cross}}, \lambda_{t+1}^{\text{join}}\}$. The overall complexity of the LARS algorithm per iteration

[1] Define $[n] \triangleq \{1, \ldots, n\}$.

is $O(nd + |\mathcal{A}|^2)$: the two searches over the active and inactive sets require $O(n|\mathcal{A}|)$ and $O(n|\mathcal{I}|)$ time, respectively, while computing the new direction $\partial\hat{\boldsymbol{\beta}}/\partial\lambda = \boldsymbol{\theta}_{t+1}$ of the regularisation path, which involves the computation of the pseudo-inverse of a submatrix of $\mathbf{X}$ specified by $\mathcal{A}$, requires $O(n|\mathcal{A}|+|\mathcal{A}|^2)$ time. When the Lasso solution is unique, it is known [Tib13] that $|\mathcal{A}| \leqslant \min\{n, d\} = n$ throughout the LARS algorithm.

1.2 Our work

In the high-dimensional setting where $d \gg n$, the search over the inactive set $\mathcal{I}$, and thus the computation of the joining time, is by far the most costly step per iteration. In this work, we propose quantum algorithms for the pathwise Lasso regression problem based on the Least Angle Regression (LARS) algorithm. To the best of our knowledge, this work is the first to present a quantum version of the LARS algorithm. We propose mainly two quantum algorithms based on the LARS algorithm to speedup the computation of the joining time. Our first quantum algorithm, called *simple quantum* LARS *algorithm*, is a straightforward improvement that utilizes the well-known quantum minimum-finding subroutine from Dürr and Høyer [DH96] to perform the search over $\mathcal{I}$. We assume that the design matrix $\mathbf{X}$ is stored in a quantum-readable read-only memory (QROM) and can be accessed in time $O(\operatorname{poly}\log(nd))$. We also assume that data can be written into quantum-readable classical-writable classical memories (QRAM) of size $O(n)$, which can be later queried in time $O(\operatorname{poly}\log n)$ [GLM08a, GLM08b]. The complexity of computing the joining time is now $O(n\sqrt{|\mathcal{I}|})$, where $O(n)$ is the time required to compute, via classical circuits, the joining times to be maximised over $\mathcal{I}$. The final runtime of our simple quantum algorithm is $\widetilde{O}(n\sqrt{|\mathcal{I}|}+n|\mathcal{A}|+|\mathcal{A}|^2)$ per iteration, where we omit polylog factors in n and d.

We then improve upon our simple quantum algorithm by approximately computing the joining times to be maximised over $\mathcal{I}$, thus reducing the factor $O(n)$ to $O(\sqrt{n})$. This is done by constructing a quantum unitary based on quantum amplitude estimation [BHMT02] that approximately computes the joining times of all the variables in the inactive set $\mathcal{I}$. We then employ the approximate quantum minimum-finding subroutine from Chen and de Wolf [CdW23] to find an estimate ϵ-close to the true maximum joining time. The result is our *approximate quantum* LARS *algorithm* that achieves a quadratic speedup in both the number of features d and the number of observations n. We also propose a sampling-based classical LARS algorithm, which can be seen as a quantisation counterpart. Our improved quantum algorithm (and the sampling-based classical algorithm), however, introduces an error in computing the joining time $\lambda_{t+1}^{\text{join}}$. As one of our main contributions, we prove that the LARS algorithm is robust to errors. More specifically, we show that, by slightly adapting the LARS algorithm, it returns (and so our approximate quantum algorithm) an *approximate regularisation path* with error proportional to the error from computing $\lambda_{t+1}^{\text{join}}$. We first define an approximate regularisation path. In the following, let $\mathbf{X}^+$ be the Moore–Penrose inverse of $\mathbf{X}$ and given a matrix $\mathbf{A}$, let $\|\mathbf{A}\|_2$ be its the spectral norm, $\|\mathbf{A}\|_1 = \max_j \sum_i |A_{ij}|$, and $\|\mathbf{A}\|_{\max} = \max_{i,j} |A_{ij}|$.

Definition 1. *Given $\epsilon : \mathbb{R}_{>0} \to \mathbb{R}_{\geqslant 0}$, we say that a vector $\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^d$ is an approximate Lasso solution with error $\epsilon(\lambda)$, or an $\epsilon(\lambda)$-minimiser, if*

$$\mathcal{L}^{(\lambda)}(\widetilde{\boldsymbol{\beta}}) - \min_{\boldsymbol{\beta}\in\mathbb{R}^d} \mathcal{L}^{(\lambda)}(\boldsymbol{\beta}) \leqslant \epsilon(\lambda), \quad \textit{where} \quad \lambda > 0.$$

A set $\widetilde{\mathcal{P}} \triangleq \{\widetilde{\boldsymbol{\beta}}(\lambda) \in \mathbb{R}^d : \lambda > 0\}$ is an approximate regularisation path with error $\epsilon(\lambda)$ if, for all $\lambda > 0$, the point $\widetilde{\boldsymbol{\beta}}(\lambda)$ of $\widetilde{\mathcal{P}}$ is an approximate Lasso solution with error $\epsilon(\lambda)$.

Result 1 (Informal version of Theorem 26)**.** *Let $T \in \mathbb{N}$, $\epsilon \geqslant 0$, and $\mathbf{X} \in \mathbb{R}^{n\times d}$. For $\mathcal{A} \subseteq [d]$ of size $|\mathcal{A}| \leqslant T$, let $\alpha_{\mathcal{A}} > 0$ be such that $\|\mathbf{X}_{\mathcal{A}}^{+}\mathbf{X}_{\mathcal{A}^c}\|_1 \leqslant \alpha_{\mathcal{A}}$. Consider an approximate* LARS *algorithm that returns a path $\widetilde{\mathcal{P}} = \{\widetilde{\boldsymbol{\beta}}(\lambda) \in \mathbb{R}^d : \lambda > 0\}$ with at most T kinks and wherein, at each iteration t, the joining time $\lambda_{t+1}^{\text{join}}$ is approximated by $\widetilde{\lambda}_{t+1}^{\text{join}}$ such that $\lambda_{t+1}^{\text{join}} \leqslant \widetilde{\lambda}_{t+1}^{\text{join}} \leqslant \left(1 - \frac{\epsilon}{1+\alpha_{\mathcal{A}}}\right)^{-1}\lambda_{t+1}^{\text{join}}$. Then $\widetilde{\mathcal{P}}$ is an approximate regularisation path with error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}(\lambda)\|_1$.*

To prove the above result, we consider an approximate version of the KKT conditions and use a duality gap for the Lasso regression. As far as we are aware, this is the first direct result on the robustness of the LARS algorithm. Compared to the work of Mairal and Yu [MY12], who also introduced an approximate LARS algorithm and employed similar techniques to analyse its correctness, the computation of the joining time in their case is exact and errors only arise by utilising a first-order optimisation method to find an approximate solution when kinks happen to be too close.

Result 1 guarantees the correctness of all our algorithms. Their time complexities, on the other hand, are given by the theorem below and summarised in Table 1.

Result 2 (Informal version of Theorems 22, 27 and 28)**.** *Let $\mathbf{X} \in \mathbb{R}^{n\times d}$, $\mathbf{y} \in \mathbb{R}^n$, $\delta \in (0,1)$, $\epsilon > 0$, and $T \in \mathbb{N}$. Assume that $\mathbf{X}$ is stored in a* QROM *and we have access to* QRAMs *and classical-samplable structures of memory size $O(n)$. For $\mathcal{A} \subseteq [d]$ of size $|\mathcal{A}| \leqslant T$, assume there are $\alpha_{\mathcal{A}}, \gamma_{\mathcal{A}} \in (0,1)$ such that*

$$\|\mathbf{X}_{\mathcal{A}}^{+}\mathbf{X}_{\mathcal{A}^c}\|_1 \leqslant \alpha_{\mathcal{A}} \qquad \text{and} \qquad \frac{\|\mathbf{X}^\top(\mathbf{I} - \mathbf{X}_{\mathcal{A}}\mathbf{X}_{\mathcal{A}}^{+})\mathbf{y}\|_\infty}{\|\mathbf{X}\|_{\max}\|(\mathbf{I} - \mathbf{X}_{\mathcal{A}}\mathbf{X}_{\mathcal{A}}^{+})\mathbf{y}\|_1} \geqslant \gamma_{\mathcal{A}},$$

where $\mathcal{A}^c = [d]\backslash\mathcal{A}$ and $\mathbf{X}_{\mathcal{A}} \in \mathbb{R}^{n\times|\mathcal{A}|}$ is the matrix formed by the columns of $\mathbf{X}$ in $\mathcal{A}$.

- *There is a quantum* LARS *algorithm that returns an optimal regularisation path with T kinks with probability at least $1-\delta$ and in time*

$$\widetilde{O}\big(n\sqrt{|\mathcal{I}|} + n|\mathcal{A}| + |\mathcal{A}|^2\big)$$

  per iteration, where $\mathcal{A}$ and $\mathcal{I}$ are the active and inactive sets of the corresponding iteration.

- *There is a quantum* LARS *algorithm that returns an approximate regularisation path with error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$ and T kinks with probability at least $1-\delta$ and in time*

$$\widetilde{O}\left(\frac{\gamma_{\mathcal{A}}^{-1} + \sqrt{n}\|\mathbf{X}\|_{\max}\|\mathbf{X}_{\mathcal{A}}^{+}\|_2}{(1-\alpha_{\mathcal{A}})\epsilon}\sqrt{|\mathcal{I}|} + n|\mathcal{A}| + |\mathcal{A}|^2\right)$$

  per iteration, where $\mathcal{A}$ and $\mathcal{I}$ are the active and inactive sets of the corresponding iteration.

- *There is a classical (dequantised)* LARS *algorithm that returns an approximate regularisation path with error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$ and T kinks with probability at least $1-\delta$ and in time*

$$\widetilde{O}\left(\frac{\gamma_{\mathcal{A}}^{-2} + n\|\mathbf{X}\|_{\max}^2\|\mathbf{X}_{\mathcal{A}}^{+}\|_2^2}{(1-\alpha_{\mathcal{A}})^2\epsilon^2}|\mathcal{I}| + n|\mathcal{A}| + |\mathcal{A}|^2\right)$$

  per iteration, where $\mathcal{A}$ and $\mathcal{I}$ are the active and inactive sets of the corresponding iteration.

The notation $\widetilde{O}(\cdot)$ omits poly log *terms in n, d, T, and δ.*

Algorithm	Error	Time complexity per iteration
Classical LARS 1	0	$O(n\lvert\mathcal{I}\rvert + n\lvert\mathcal{A}\rvert + \lvert\mathcal{A}\rvert^2)$
Simple quantum LARS 2	0	$\widetilde{O}(n\sqrt{\lvert\mathcal{I}\rvert} + n\lvert\mathcal{A}\rvert + \lvert\mathcal{A}\rvert^2)$
Approximate classical LARS 4	$\lambda\epsilon\lVert\widetilde{\boldsymbol{\beta}}\rVert_1$	$\widetilde{O}\left(\frac{\gamma_{\mathcal{A}}^{-2}+n\lVert\mathbf{X}\rVert_{\max}^2\lVert\mathbf{X}_{\mathcal{A}}^+\rVert_2^2}{(1-\alpha_{\mathcal{A}})^2\epsilon^2}\lvert\mathcal{I}\rvert + n\lvert\mathcal{A}\rvert + \lvert\mathcal{A}\rvert^2\right)$
Approximate quantum LARS 3	$\lambda\epsilon\lVert\widetilde{\boldsymbol{\beta}}\rVert_1$	$\widetilde{O}\left(\frac{\gamma_{\mathcal{A}}^{-1}+\sqrt{n}\lVert\mathbf{X}\rVert_{\max}\lVert\mathbf{X}_{\mathcal{A}}^+\rVert_2}{(1-\alpha_{\mathcal{A}})\epsilon}\sqrt{\lvert\mathcal{I}\rvert} + n\lvert\mathcal{A}\rvert + \lvert\mathcal{A}\rvert^2\right)$

Table 1: Summary of results. Throughout this work, n is the number of observations, d is the number of features, $\mathcal{A}$ is the active set, and $\mathcal{I}$ is the inactive set. In addition, $\epsilon > 0$ is the approximation error, $\delta \in (0,1)$ is the failure probability, T is the number of kinks in the regularisation path $\mathcal{P}$, $\mathbf{X}_{\mathcal{A}} = [\mathbf{X}_i]_{i\in\mathcal{A}} \in \mathbb{R}^{n\times|\mathcal{A}|}$ is the matrix formed by the columns of $\mathbf{X}$ in $\mathcal{A}$, $\mathbf{X}_{\mathcal{A}}^+$ is the Moore–Penrose inverse of $\mathbf{X}_{\mathcal{A}}$, and the parameters $\alpha_{\mathcal{A}}, \gamma_{\mathcal{A}} \in (0,1)$ are defined in Result 2. The notation $\widetilde{O}(\cdot)$ omits $\operatorname{poly}\log$ terms in n, d, T, and δ.

Our approximate LARS algorithms output a regularisation path with point-wise error $\lambda\epsilon\lVert\widetilde{\boldsymbol{\beta}}\rVert_1$. Even though this is not a purely additive error ϵ, it is still better than a purely multiplicative error $\epsilon\mathcal{L}^{(\lambda)}(\widetilde{\boldsymbol{\beta}})$. It is possible to prove that $\mathcal{L}^{(\lambda)}(\widetilde{\boldsymbol{\beta}}) \leqslant \frac{1}{1-\epsilon}\min_{\boldsymbol{\beta}\in\mathbb{R}^d}\mathcal{L}^{(\lambda)}(\boldsymbol{\beta})$ (see Remark 25), therefore our approximate LARS algorithms output a solution with error at most $\frac{\epsilon}{1-\epsilon}\min_{\boldsymbol{\beta}\in\mathbb{R}^d}\mathcal{L}^{(\lambda)}(\boldsymbol{\beta})$. Moreover, this type of error $\lambda\epsilon\lVert\widetilde{\boldsymbol{\beta}}\rVert_1$ is standard in the literature, see Section 6 and also [BvdG11, MY12, Wai19].

The complexity of our approximate quantum LARS algorithm depends on a few properties of the design matrix $\mathbf{X}$: the quantity $\lVert\mathbf{X}\rVert_{\max}\lVert\mathbf{X}_{\mathcal{A}}^+\rVert_2$ and the parameters $\alpha_{\mathcal{A}}$ and $\gamma_{\mathcal{A}}$. We note that $\lVert\mathbf{X}\rVert_{\max}\lVert\mathbf{X}_{\mathcal{A}}^+\rVert_2 \leqslant \lVert\mathbf{X}\rVert_2\lVert\mathbf{X}^+\rVert_2$ is upper bounded by the condition number of $\mathbf{X}$, which is simply a constant for well-behaved matrices. The existence of the parameter $\alpha_{\mathcal{A}} \in [0,1)$ is often called *(mutual) incoherence* or *strong irrepresentable condition* in the literature [ZY06, HMZ08, Wai19] and is related to the level of orthogonality between columns of $\mathbf{X}$: if the column space of $\mathbf{X}_{\mathcal{A}}$ is orthogonal to $\mathbf{X}_i$, then $\lVert\mathbf{X}_{\mathcal{A}}^+\mathbf{X}_i\rVert_1 = 0$. Mutual incoherence has been considered by several previous works [DH06, EB02, Fuc04, Tro06, ZY06, MB06, Wai09]. Refs. [ZY06, MB06] provide several examples of families of matrices that satisfy mutual incoherence. Finally, the parameter $\gamma_{\mathcal{A}}$, introduced and named by us as *mutual overlap (between* $\mathbf{y}$ *and* $\mathbf{X}$*)*, measures the overlap between projections of the observation vector $\mathbf{y}$ and the columns of $\mathbf{X}$ (note that $\gamma_{\mathcal{A}} \leqslant 1$ by Hölder's inequality).

In the case when $\lVert\mathbf{X}\rVert_{\max}\lVert\mathbf{X}_{\mathcal{A}}^+\rVert_2 = O(1)$ and $\alpha_{\mathcal{A}}$ and $\gamma_{\mathcal{A}}$ are bounded away from 1 and 0, respectively, the complexity per iteration is $\widetilde{O}(\sqrt{n|\mathcal{I}|}/\epsilon + n|\mathcal{A}| + |\mathcal{A}|^2)$. For $\epsilon = 1/\operatorname{poly}\log d$ and iterations when $|\mathcal{A}| = O(n)$, we obtain the overall quadratic improvement $\widetilde{O}(\sqrt{nd})$ over the classical LARS algorithm. A further speedup beyond quadratic can be obtained if $\lVert\mathbf{X}\rVert_{\max}\lVert\mathbf{X}_{\mathcal{A}}^+\rVert_2 = o(1)$. Our approximate quantum LARS algorithm, however, depends on the design matrix being well behaved in order to bound the three quantities mentioned above. In Section 4.4, we bound $\lVert\mathbf{X}\rVert_{\max}\lVert\mathbf{X}_{\mathcal{A}}^+\rVert_2$, $\alpha_{\mathcal{A}}$, and $\gamma_{\mathcal{A}}$ for the case when $\mathbf{X}$ is a random matrix with rows sampled i.i.d. from the multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with covariance matrix $\boldsymbol{\Sigma} \succ \mathbf{0} \in \mathbb{R}^{d\times d}$.

Result 3 (Informal version of Lemma 29 to 31)**.** *Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be a random matrix with rows sampled i.i.d. from the Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with positive-definite covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$. Let $\mathcal{A} \subseteq [d]$ of size $|\mathcal{A}| = O(n/\log d)$. If $\boldsymbol{\Sigma}$ is "well behaved" (which includes $\boldsymbol{\Sigma} = \mathbf{I}$), then, with high probability,*

$$\lVert\mathbf{X}\rVert_{\max}\lVert\mathbf{X}_{\mathcal{A}}^+\rVert_2 = O\left(\sqrt{\frac{\log d}{n}}\right), \qquad \alpha_{\mathcal{A}} = 1 - \Omega(1), \qquad \text{and} \qquad \gamma_{\mathcal{A}} = \Omega\left(\frac{\lVert\mathbf{y}\rVert_2/\lVert\mathbf{y}\rVert_1}{\sqrt{\log d}}\right).$$

As a consequence, the approximate quantum and classical LARS *algorithms output an approximate regularisation path with error $\lambda\epsilon\lVert\widetilde{\boldsymbol{\beta}}\rVert_1$ and $T = O(n/\log d)$ kinks with probability at least $1-\delta$ in*

time $\widetilde{O}((\|\mathbf{y}\|_1/\|\mathbf{y}\|_2)(\sqrt{d}/\epsilon))$ *and* $\widetilde{O}((\|\mathbf{y}\|_1/\|\mathbf{y}\|_2)(d/\epsilon^2))$ *per iteration, respectively, up to* poly log *term in* n*,* d*, and* δ*. If further* $\|\mathbf{y}\|_1/\|\mathbf{y}\|_2 = \operatorname{poly}\log n$*, then the quantum and classical complexities per iteration are* $\widetilde{O}(\sqrt{d}/\epsilon)$ *and* $\widetilde{O}(d/\epsilon^2)$*, respectively.*

For the exact meaning of a "well-behaved" covariance matrix $\mathbf{\Sigma}$, we refer the reader to Section 4.4. On a high level, it means that quantities like the minimum singular value of $\mathbf{\Sigma}$, the norm $\|\sqrt{\mathbf{\Sigma}}\|_1$, and other related properties of the conditional covariance matrix of $\mathbf{X}_{\mathcal{A}}|\mathbf{X}_{\mathcal{A}^c}$ are constant. Mostly special, the case $\mathbf{\Sigma} = \mathbf{I}$, which corresponds to a standard Gaussian matrix with entries sampled i.i.d. from $\mathcal{N}(0,1)$, trivially fulfills all these requirements.

Result 3 implies that the complexity of our approximate LARS algorithms, both quantum and classical, can be substantially better than the standard LARS algorithm depending on the input data. In the case when $\mathbf{X}$ is a Gaussian matrix and $\mathbf{y}$ is sparse, so that $\|\mathbf{y}\|_1/\|\mathbf{y}\|_2 = \operatorname{poly}\log n$, the approximate quantum and classical complexities per iteration are $\widetilde{O}(\sqrt{d}/\epsilon)$ and $\widetilde{O}(d/\epsilon^2)$, respectively, which only depend polylogarithmically on the number of samples n (assuming that $n|\mathcal{A}| = O(\sqrt{d})$). The quantum runtime, in particular, is more than quadratically better than the usual $O(nd)$ classical runtime (at the expense of approximating the solution). The condition that $\mathbf{y}$ is sparse can be artificially generated through setting to zero some of the regression coefficients of the underlying true solution $\boldsymbol{\beta}^*$ connecting $\mathbf{X}$ and $\mathbf{y}$ (see Section 6). We note that sparse inputs have also provided speedups in other settings [BACS07, HHL09, BCC+14, Pra14, LGZ16, GSLW19, LDR23]. Independently, we believe that this result would be of interest to the compressed sensing community. The complexities for the case of random Gaussian design matrix $\mathbf{X}$ are summarised in Table 2.

Finally, we prove query lower bounds for any classical and quantum algorithms for Lasso. The proof techniques come mostly from the work of Chen and de Wolf [CdW23] but adapted to the Lasso problem from Equation (1) instead of $\min_{\boldsymbol{\beta}\in\mathbb{R}^n} \frac{1}{n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$ such that $\|\boldsymbol{\beta}\|_1 \leqslant 1$.

Result 4 (Theorems 43 and 44)**.** *Let* $n < d$ *such that* $n = \Omega(\log d)$*,* $\lambda = \Omega(n)$*, and* $\epsilon \in (0, 1/5)$ *such that* $\epsilon = \Omega(\sqrt{n^3 \log d}/\lambda^2)$*. There is* $\mathbf{X} \in \{-1,1\}^{n\times d}$ *such that every bounded-error classical or quantum algorithm that computes an* $\epsilon\lambda\|\widetilde{\boldsymbol{\beta}}\|_1$*-minimiser for the Lasso* $\mathcal{L}^{(\lambda)}(\boldsymbol{\beta})$ *uses*

- *classical:* $\Omega\left(\frac{n^4 d}{\lambda^4\epsilon^2 \log d}\right)$ *queries to* $\mathbf{X}$*. For* $\epsilon = \Theta(\sqrt{n^3 \log d}/\lambda^2)$*, this is* $\Omega\left(\frac{nd}{\log^2 d}\right)$*;*
- *quantum:* $\Omega\left(\frac{n^3\sqrt{d}}{\lambda^3\epsilon^{3/2}}\right)$ *queries to* $\mathbf{X}$*. For* $\epsilon = \Theta(\sqrt{n^3 \log d}/\lambda^2)$*, this is* $\Omega\left(\frac{n^{3/4}\sqrt{d}}{\log^{3/4} d}\right)$*.*

We stress that our lower bound does not contradict the results from Table 2 since it is a worst-case bound: there is a design matrix $\mathbf{X}$ such that any classical or quantum algorithm requires $\Omega\left(\frac{n^4 d}{\lambda^4\epsilon^2 \log d}\right)$ or $\Omega\left(\frac{n^3\sqrt{d}}{\lambda^3\epsilon^{3/2}}\right)$ queries to $\mathbf{X}$, respectively, which does not exclude the existence of easy instances, a trivial case being the all-0 matrix. Still, for the range of parameters $\lambda = \Theta(n)$ and $\epsilon = \Theta(\sqrt{n^3 \log d}/\lambda^2) = \Theta(1)$, the time complexity per iteration of our approximate quantum LARS algorithm with a random Gaussian matrix is $\widetilde{O}(\sqrt{d})$, while for the approximate classical LARS it is $\widetilde{O}(d)$. Together with $O(n)$ iterations in order to reach the desired λ, their final complexities would be $\widetilde{O}(nd)$ (classical) and $\widetilde{O}(n\sqrt{d})$ (quantum), larger than their respective lower bounds $\widetilde{\Omega}(nd)$ and $\widetilde{\Omega}(n^{3/4}\sqrt{d})$. We note that, in this case, the time complexity of the usual classical LARS algorithm is $O(n^2 d)$ (although the solution has no error).

Compared to the previous work of Chen and de Wolf [CdW23] on quantum algorithms for Lasso, our work is different in that we study the pathwise Lasso regression problem as the tuning parameter λ varies, and not focus on a single point λ. It looks infeasible to retrieve even a small interval $[\hat{\boldsymbol{\beta}}(\lambda), \hat{\boldsymbol{\beta}}(\lambda')]$ of Lasso solutions via their algorithms. On the other hand, if one wishes to compute $\hat{\boldsymbol{\beta}}(\lambda)$ for a specific λ, then their algorithms would be preferable. Both works are thus incomparable in

Algorithm	Error	Time complexity per iteration
Classical LARS 1	0	$O(nd)$
Simple quantum LARS 2	0	$\widetilde{O}(n\sqrt{d} + n\lvert\mathcal{A}\rvert)$
Approximate classical LARS 4	$\lambda\epsilon\lVert\widetilde{\boldsymbol{\beta}}\rVert_1$	$\widetilde{O}(d/\epsilon^2 + n\lvert\mathcal{A}\rvert)$
Approximate quantum LARS 3	$\lambda\epsilon\lVert\widetilde{\boldsymbol{\beta}}\rVert_1$	$\widetilde{O}(\sqrt{d}/\epsilon + n\lvert\mathcal{A}\rvert)$

Table 2: Summary of results for the case when the design matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ is a random Gaussian matrix with rows sampled i.i.d. from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with well-behaved covariance matrix $\boldsymbol{\Sigma} \succ \mathbf{0} \in \mathbb{R}^{d\times d}$ and the vector of observations $\mathbf{y} \in \mathbb{R}^n$ is such that $\lVert\mathbf{y}\rVert_1/\lVert\mathbf{y}\rVert_2 = \operatorname{poly}\log n$. These results are valid for active and inactive sets $\mathcal{A}$ and $\mathcal{I}$ of size $\lvert\mathcal{A}\rvert = O(n/\log d)$ and $\lvert\mathcal{I}\rvert = \Theta(d)$. The notation $\widetilde{O}(\cdot)$ omits $\operatorname{poly}\log$ terms in n, d, and the failure probability δ.

terms of results. The output solution from the quantum algorithm of Chen and de Wolf can actually be used to initialise our algorithms instead of the point $(\lambda_0, \hat{\boldsymbol{\beta}}(\lambda_0) = \mathbf{0})$. Regarding quantum techniques, both works employ similar quantum subroutines like quantum amplitude estimation and quantum minimum finding. The fact that no advanced subroutines are needed to improve the dependence on both n and d should be seen as a strength of our (and [CdW23]) work.

The paper is organised as follows. In Section 2, we introduce important notations, the computational model, and useful quantum subroutines. In Section 3, we describe the steps to obtain a Lasso path, conditions for the uniqueness of the Lasso solution, and analyse the per-iteration time complexity of the classical LARS algorithm. In Section 4, we describe our two quantum LARS algorithms and a classical equivalent algorithm based on sampling. We discuss bounds in the noisy regime in Section 6 and summarise our work with possible future directions in Section 7.

1.3 Related work

Seeing the importance of variable selection in the high-dimensional data context, algorithms such as best subsets, forward selection, and backward elimination are well studied and widely used to produce sparse solutions [Mao02, Mao04, WFL00, VK05, BT19, RS14, TFZB08, ZJ96, KM14, WB07]. Apart from the aforementioned feature selection algorithms, Ridge regression, proposed by Hoerl and Kennard [HK70], is another popular and well-studied method [McD09, Dor14, HK70, MS75, vW15, HKB75, SGV98, Kib03, Vin78].

The homotopy method of Osborne *et al.* [OPT00a, OPT00b] and the LARS algorithm of Efron *et al.* [EHJT04] are feature selection algorithms which use a less greedy approach as compared to classical forward selection approaches. The LARS algorithm can be modified to give rise to algorithms for Lasso and for the efficient implementation of Forward Stagewise linear regression. Based on the work of Efron *et al.* [EHJT04], Rosset and Zhu [RZ07] gave a general characterization of loss-penalty pairs to allow for efficient generation of the full regularised coefficient paths. Mairal and Yu [MY12] later provided an ϵ-approximate path for Lasso with at most $O(1/\sqrt{\epsilon})$ segments can be obtained by developing an approximate homotopy algorithm based on newly defined ϵ-approximate optimality conditions. Other works on algorithms for Lasso and its variants include Refs. [Rot04, Tib13, Sto13, KKK08, GKZ18, AT16].

In the quantum setting, considerable amount of work has been done on linear regression using the (ordinary/unregularised) least squares approach. Kaneko *et al.* [KMTY21] proposed a quantum algorithm for linear regression which outputs classical estimates of the regression coefficients. This improves upon the earlier works [WBL12, Wan17] where the regression coefficients are encoded in the amplitudes of a quantum state. Chakraborty *et al.* [CGJ19] devised quantum algorithms for the weighted and generalised variants of the least squares problem by using block-encoding techniques

and Hamiltonian simulation [LC19] to improve the quantum liner systems solver. As an application of linear regression to machine learning tasks, Schuld *et al.* [SSP16] designed a quantum pattern recognition algorithm based on the ordinary least squares approach, using ideas from [HHL09, LMR14]. Kerenidis and Prakash [KP20] provided a quantum stochastic gradient descent algorithm for the weighted least squares problem using a quantum linear systems solver in the QRAM data structure model [Pra14, KP17], which allows for efficient quantum state preparation.

While quantum linear regression has been relatively well studied in the unregularised setting, research on algorithms for regression using regularised least squares is not yet well established. In the sparse access model, Yu *et al.* [YGW21] presented a quantum algorithm for Ridge regression using a technique called parallel Hamiltonian simulation to develop the quantum analogue of K-fold cross-validation [HK70], which serves as an efficient estimator of Ridge regression. Other works on quantum Ridge regression algorithms in the sparse access model include [SX20, Sha23, CYGL23]. Chen and de Wolf [CdW23] designed quantum algorithms for Lasso and Ridge regression from the perspective of empirical loss minimisation. Both of their quantum algorithms output a classical vector whose loss is ϵ-close to the minimum achievable loss. Around the same time, Bellante and Zanero [BZ22, Bel24] gave a polynomial speedup for the classical matching-pursuit algorithm, which is a heuristic algorithm for the best subset selection model. More recently, Chakraborty *et al.* [CMP23] gave the first quantum algorithms for least squares with general ℓ_2-norm regularisation, which includes regularised versions of quantum ordinary least squares, quantum weighted least squares, and quantum generalised least squares. Their algorithms use block-encoding techniques and the framework of quantum singular value transformation [GSLW19], and demonstrate substantial improvement compared to previous results on quantum Ridge regression [SX20, CYGL23, YGW21].

2 Preliminaries

2.1 Notations

For a positive integer $n \in \mathbb{N}$, let $[n] \triangleq \{1, \dots, n\}$. Given a vector $\mathbf{u} \in \mathbb{R}^n$, we denote its i-th entry as u_i for $i \in [n]$. Let $\|\mathbf{u}\|_p \triangleq \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$ and by $\mathcal{D}_\mathbf{u}$ we denote the distribution over $[n]$ with probability density function $\mathcal{D}_\mathbf{u}(i) = |u_i|/\|\mathbf{u}\|_1$. Given a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, we denote its i-th column as $\mathbf{X}_i$ for $i \in [d]$. Let $\mathbf{I}$ be the identity matrix. Given $\mathcal{A} = \{i_1, \dots, i_k\} \subseteq [d]$, define $\mathcal{A}^c = [d]\backslash\mathcal{A}$ and let $\mathbf{X}_\mathcal{A} \in \mathbb{R}^{n\times|\mathcal{A}|}$ be the matrix $\mathbf{X}_\mathcal{A} = [\mathbf{X}_{i_1}, \dots, \mathbf{X}_{i_k}]$ formed by the columns of $\mathbf{X}$ in $\mathcal{A}$. The previous notation naturally extends to vectors, i.e., $\mathbf{u}_\mathcal{A} = [u_{i_1}, \dots, u_{i_k}]$. Given $\mathcal{A} \subseteq [d]$ and $S \subseteq [n]$, $\mathbf{X}_{S\mathcal{A}}$ is the submatrix of $\mathbf{X}$ with rows in S and columns in $\mathcal{A}$. We denote by $\mathrm{col}(\mathbf{X})$, $\mathrm{row}(\mathbf{X})$, $\mathrm{null}(\mathbf{X})$, and $\mathrm{rank}(\mathbf{X})$ the column space, row space, null space, and the rank, respectively, of $\mathbf{X}$. We denote by $\mathbf{X}^+$ the Moore-Penrose inverse of $\mathbf{X}$. If $\mathbf{X}$ has linearly independent columns, $\mathbf{X}^+ = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$. Note that $\mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+$ is the orthogonal projector onto $\mathrm{col}(\mathbf{X}_\mathcal{A})$. Let $\|\mathbf{X}\|_p \triangleq \sup_{\mathbf{u}\in\mathbb{R}^d:\|\mathbf{u}\|_p=1} \|\mathbf{X}\mathbf{u}\|_p$ be the p-norm of $\mathbf{X}$, $p \in [1, \infty]$. As special cases, $\|\mathbf{X}\|_1 = \max_{j\in[d]} \sum_{i=1}^n |X_{ij}|$ and $\|\mathbf{X}\|_\infty = \max_{i\in[n]} \sum_{j=1}^d |X_{ij}|$. Let also $\|\mathbf{X}\|_{\max} \triangleq \max_{i\in[n],j\in[d]} |X_{ij}|$. We use $|\bar{0}\rangle$ to denote the state $|0\rangle \otimes \cdots \otimes |0\rangle$, where the number of qubits is clear from the context. We use $\widetilde{O}(\cdot)$ to hide polylogarithmic factors, i.e., $\widetilde{O}(f(n)) = O(f(n) \cdot \mathrm{poly}\log(f(n)))$. See Appendix A for a summary of symbols.

2.2 Concentration bounds

We revise a few notions of probability theory and concentration bounds. Let $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ be the multivariate Gaussian distribution with covariance matrix $\mathbf{\Sigma} > \mathbf{0} \in \mathbb{R}^{d\times d}$. We say that a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ is drawn from the $\mathbf{\Sigma}$-Gaussian ensemble if the rows of $\mathbf{X}$ are sampled i.i.d. according to

$\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. We note that $\mathbf{Z}$ is drawn from the $\mathbf{I}$-Gaussian ensemble if and only if $\mathbf{X} = \mathbf{Z}\sqrt{\mathbf{\Sigma}}$ is drawn from the $\mathbf{\Sigma}$-Gaussian ensemble. We shall use the concept of sub-Gaussian variable.

Definition 2 (Sub-Gaussianity)**.** *A random single-variable z is sub-Gaussian with parameter σ^2 if $\mathbb{E}[e^{t(z-\mathbb{E}[z])}] \leqslant \exp\left(\frac{t^2\sigma^2}{2}\right)$ for all $t \in \mathbb{R}$.*

We note that any sub-Gaussian random variable satisfies a Chernoff bound.

Fact 3. *If z is a sub-Gaussian random variable with parameter σ^2, then*

$$\mathbb{P}[z - \mathbb{E}[z] > t] \leqslant \exp\left(-\frac{t^2}{2\sigma^2}\right) \quad \text{and} \quad \mathbb{P}[|z - \mathbb{E}[z]| > t] \leqslant 2\exp\left(-\frac{t^2}{2\sigma^2}\right) \quad \text{for all } t \in \mathbb{R}.$$

Fact 4. *If each coordinate of $\mathbf{w} \in \mathbb{R}^n$ is an independent sub-Gaussian random variable with parameter σ^2, then, for any $\mathbf{u} \in \mathbb{R}^n$, $\mathbf{u}^\top\mathbf{w}$ is sub-Gaussian with parameter $\sigma^2\|\mathbf{u}\|_2^2$.*

Proof. $\mathbb{E}\left[\exp\left(t\,\mathbf{u}^\top\mathbf{w}\right)\right] = \prod_{i=1}^n \mathbb{E}\left[\exp\left(t\,u_i w_i\right)\right] \leqslant \prod_{i=1}^n \exp\left(\frac{t^2\sigma^2u_i^2}{2}\right) = \exp\left(\frac{t^2\sigma^2\|\mathbf{u}\|_2^2}{2}\right)$, $\forall t \in \mathbb{R}$. □

More generally, the sum of bounded independent random variables can be bounded via Hoeffding's inequality.

Fact 5 (Hoeffding's bound)**.** *Let $z_1, \ldots, z_n$ be independent random variables such that, for $i \in [n]$, $a_i \leqslant z_i \leqslant b_i$ almost surely. Let $z \triangleq \sum_{i=1}^n z_i$. Then*

$$\mathbb{P}[|z - \mathbb{E}[z]| \geqslant t] \leqslant 2\exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right) \quad \text{for all } t > 0.$$

Lévy's lemma [MS86, Led01, AGZ09, Ver18, Mec19] will also be useful.

Fact 6 (Lévy's lemma, [AGZ09, Corollary 4.4.28])**.** *Let $f : \mathbb{S}^d \to \mathbb{R}$ be a function defined on the d-dimensional hypersphere $\mathbb{S}^d$. Assume f is K-Lipschitz, i.e., $|f(\psi) - f(\phi)| \leqslant K\|\psi - \phi\|$. Then*

$$\mathop{\mathbb{P}}_{\psi\sim\mu_{\text{Haar}}}[|f(\psi) - \mathbb{E}[f]| \geqslant \epsilon] \leqslant 2\exp\left(-\frac{d\epsilon^2}{4K^2}\right) \quad \text{for all } \epsilon > 0.$$

We will also make use of the following lemma about the relative approximation of a ratio of two real numbers that are given by relative approximations.

Lemma 7. *Let $\widetilde{a}, \widetilde{b} \in \mathbb{R}$ be estimates of $a, b \in \mathbb{R}\backslash\{0\}$, respectively, such that $|a - \widetilde{a}| \leqslant \epsilon_a$ and $|b - \widetilde{b}| \leqslant \epsilon_b$, where $\epsilon_a \geqslant 0$ and $\epsilon_b \in [0, |b|/2]$. Then*

$$\left|\frac{\widetilde{a}}{\widetilde{b}} - \frac{a}{b}\right| \leqslant 2\frac{|a|}{|b|}\left(\frac{\epsilon_a}{|a|} + \frac{\epsilon_b}{|b|}\right).$$

Proof. First note that $|b| - |\widetilde{b}| \leqslant \epsilon_b \implies \frac{1}{|\widetilde{b}|} \leqslant \frac{1}{|b|-\epsilon_b} \leqslant \frac{2}{|b|}$. Then

$$\left|\frac{\widetilde{a}}{\widetilde{b}} - \frac{a}{b}\right| = \left|\frac{\widetilde{a}b - ab + ab - a\widetilde{b}}{b\widetilde{b}}\right| \leqslant \left|\frac{\widetilde{a} - a}{\widetilde{b}}\right| + \left|\frac{a(b - \widetilde{b})}{b\widetilde{b}}\right| \leqslant \frac{2\epsilon_a}{|b|} + \frac{|a|}{|b|}\frac{2\epsilon_b}{|b|} = 2\frac{|a|}{|b|}\left(\frac{\epsilon_a}{|a|} + \frac{\epsilon_b}{|b|}\right). \quad \square$$

2.3 Computational model

Classical computational model. Our classical computational model is a classical random-access machine. The input to the Lasso problem is a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and a vector $\mathbf{y} \in \mathbb{R}^n$, which are stored in a classical-readable read-only memory (ROM). For simplicity, reading any entry of $\mathbf{X}$ and $\mathbf{y}$ takes constant time. The classical computer can write bits to a classical-samplable-and-writable memory. More specifically, the classical computer can write a vector $\mathbf{u} \in \mathbb{R}^m$ (or parts of it) into a low-overhead samplable data structure in time $O(m)$, which allows it to sample an index $i \in [m]$ from the probability distribution $\mathcal{D}_{\mathbf{u}}$ in time $O(\operatorname{poly}\log m)$. We refer to the ability to sample from $\mathcal{D}_{\mathbf{u}}$ as having sampling access to the vector $\mathbf{u}$. In this work, all classical-samplable-and-writable memories have size $m = O(n)$. We assume an arithmetic model in which we ignore issues arising from the fixed-point representation of real numbers. All basic arithmetic operations in this model take constant time.

Quantum computational model. Our quantum computational model is a classical random-access machine with access to a quantum computer. The input to the Lasso problem is a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and a vector $\mathbf{y} \in \mathbb{R}^n$, which are stored in a classical-readable read-only memory (ROM). For simplicity, reading any entry of $\mathbf{X}$ and $\mathbf{y}$ takes constant time. We assume that $\mathbf{X}$ is also stored in a quantum-readable read-only memory (QROM), whose single entries can be queried. This means that the quantum computer has access to an oracle $\mathcal{O}_{\mathbf{X}}$ that performs the mapping

$$\mathcal{O}_{\mathbf{X}} : |i,j\rangle|\bar{0}\rangle \mapsto |i,j\rangle|X_{ij}\rangle, \quad \forall i \in [n], j \in [d],$$

in time $O(\operatorname{poly}\log(nd))$. The classical computer can also write bits to a quantum-readable classical-writable classical memory (QRAM) [GLM08a, GLM08b, ABD+24], which can be accessed in superposition by the quantum computer. More specifically, the classical computer can write a vector $\mathbf{u} \in \mathbb{R}^m$ (or parts of it) into the memory of a QRAM in time $O(m)$, which allows the quantum computer to invoke an oracle $\mathcal{U}_{\mathbf{u}}$ that performs the mapping

$$\mathcal{U}_{\mathbf{u}} : |i\rangle|\bar{0}\rangle \mapsto |i\rangle|u_i\rangle, \quad \forall i \in [m],$$

in time $O(\operatorname{poly}\log m)$. We refer to the ability to invoke $\mathcal{U}_{\mathbf{u}}$ as having quantum access to the vector $\mathbf{u}$. In this work, all QRAMs have size $m = O(n)$. We do not assume that $\mathbf{y}$ has been previously stored in a QROM. Instead, it can be stored in a QRAM in time $O(n)$ by the classical computer.

The classical computer can send the description of a quantum circuit to the quantum computer, which is a sequence of quantum gates from a universal gate set plus queries to the bits stored in the QROM and/or QRAM; the quantum computer runs the circuit, performs a measurement in the computational basis, and returns the measurement outcome to the classical computer. We refer to the runtime of a classical/quantum computation as the number of basic gates performed plus the time complexity of all the calls to the QROM and QRAMs. We assume an arithmetic model in which we ignore issues arising from the fixed-point representation of real numbers. All basic arithmetic operations in this model take constant time.

Kerenidis and Prakash [Pra14, KP17] introduced a classical data structure (sub-sequentially called KP-tree) to store a vector $\mathbf{u} \in \mathbb{R}^m$ to enable the efficient preparation of the state

$$\sum_{i=1}^{m} \sqrt{\frac{|u_i|}{\|\mathbf{u}\|_1}} |i\rangle|\operatorname{sign}(u_i)\rangle.$$

We shall commonly use KP-trees of auxiliary vectors in order to prepare the above states.

Fact 8 (KP-tree [Pra14, KP17, CdW23])**.** *Let $\mathbf{u} \in \mathbb{R}^m$ be a vector with $w \in \mathbb{N}$ non-zero entries. There is a data structure called* KP*-tree of size $O(w \operatorname{poly}\log m)$ that stores each input (i, u_i) in time $O(\operatorname{poly}\log m)$. After the data structure has been constructed, there is a quantum algorithm that prepares the state $\sum_{i=1}^m \sqrt{\frac{|u_i|}{\|\mathbf{u}\|_1}}|i\rangle|\operatorname{sign}(u_i)\rangle$ up to negligible error in time $O(\operatorname{poly}\log m)$.*

Remark 9. *It can be argued that the* QROM/QRAM *model is too strong of an assumption and it might not be realistic* [JR23]*. Nonetheless, we believe it is a natural and fair quantum equivalent to the classical computational model. If we make the assumption that the input can be classically accessed in constant time and/or sampled in poly-logarithmically time, then a quantum computer should also require, in all fairness, a comparable amount of time to quantumly access the input.*

2.4 Quantum subroutines

In this section, we review some useful quantum subroutines that shall be used in the rest of the paper, starting with the quantum minimum-finding algorithm from Dürr and Høyer [DH96].

Fact 10 (Quantum minimum finding [DH96])**.** *Given quantum access to a vector $\mathbf{u} \in \mathbb{R}^m$, there is a quantum algorithm that finds $\min_{i\in[m]} u_i$ with success probability $1-\delta$ in time $\widetilde{O}(\sqrt{m}\log\frac{1}{\delta})$.*

The above subroutine was later generalised by Chen and de Wolf [CdW23] in the case when one has quantum access to the entries of $\mathbf{u}$ up to some additive error. We note that a similar result, but in a different setting, appeared in [vAGGdW17, QCR20].

Fact 11 ([CdW23, Theorem 2.4])**.** *Let $\delta_1, \delta_2 \in (0,1)$ such that $\delta_2 = O\big(\delta_1^2/(m\log(1/\delta_1))\big)$, $\epsilon > 0$, and $\mathbf{u} \in \mathbb{R}^m$. Suppose access to a unitary that maps $|k\rangle|\bar{0}\rangle \mapsto |k\rangle|R_k\rangle$ such that, for every $k \in [m]$, after measuring the state $|R_k\rangle$, with probability at least $1-\delta_2$ the first register r_k of the measurement outcome satisfies $|r_k - u_k| \leqslant \epsilon$. Then there is a quantum algorithm that finds an index $k \in [m]$ such that $u_k \leqslant \min_{j\in[m]} u_j + 2\epsilon$ with probability at least $1-\delta_1$ and in time $\widetilde{O}(\sqrt{m}\log(1/\delta_1))$.*

We shall also need the amplitude estimation subroutine from Brassard *et al.* [BHMT02].

Fact 12 ([BHMT02, Theorem 12])**.** *Given $M \in \mathbb{N}$ and access to an $(n+1)$-qubit unitary $\mathbf{U}$ satisfying*

$$\mathbf{U}|0^n\rangle|0\rangle = \sqrt{a}|\psi_1\rangle|1\rangle + \sqrt{1-a}|\psi_0\rangle|0\rangle,$$

where $|\psi_0\rangle$ and $|\psi_1\rangle$ are arbitrary n-qubit states and $a \in (0,1)$, there is a quantum algorithm that uses $O(M)$ applications of $\mathbf{U}$ and $\mathbf{U}^\dagger$ and $\widetilde{O}(M)$ elementary gates, and outputs a state $|\Lambda\rangle$ such that, after measuring $|\Lambda\rangle$, with probability at least $9/10$, the first register λ of the outcome satisfies

$$|a-\lambda| \leqslant \frac{\sqrt{a(1-a)}}{M} + \frac{1}{M^2}.$$

Finally, the following result, which is based on [CdW23, Theorem 3.4], constructs the mapping $|j\rangle|\bar{0}\rangle \mapsto |j\rangle|R_j\rangle$ such that R_j holds an approximation to $\mathbf{A}_j^\top\mathbf{u}$ up to an additive error, where $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $\mathbf{u} \in \mathbb{R}^n$ can be accessed quantumly via KP-trees.

Lemma 13. *Given $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $\mathbf{u} \in \mathbb{R}^n$, assume they are stored in* KP*-trees and we have quantum access to their entries. Assume that $\|\mathbf{u}\|_\infty$ and $\|\mathbf{A}_j\|_\infty$, $j \in [d]$, are known. Let $\delta \in (0,1)$ and $\epsilon > 0$. Then there is a quantum algorithm that implements the mapping $|j\rangle|\bar{0}\rangle \mapsto |j\rangle|R_j\rangle$ such that, for all $j \in [d]$, after measuring the state $|R_j\rangle$, with probability at least $1-\delta$, the outcome r_j of the first register satisfies*

$$\left|r_j - \mathbf{A}_j^\top\mathbf{u}\right| \leqslant \epsilon \min(\|\mathbf{A}_j\|_\infty\|\mathbf{u}\|_1, \|\mathbf{A}_j\|_1\|\mathbf{u}\|_\infty)$$

in time $O(\epsilon^{-1}\log(1/\delta)\operatorname{poly}\log(nd))$.

Proof. Fix $j \in [d]$. We start by approximating $\mathbf{A}_j^\top \mathbf{u} = \sum_{i=1}^n A_{ij} u_i$ up to error $\epsilon \|\mathbf{A}_j\|_\infty \|\mathbf{u}\|_1$. Apply the operator from Fact 8 to create

$$\sum_{i=1}^{n} \sqrt{\frac{|u_i|}{\|\mathbf{u}\|_1}} |i\rangle |s_i\rangle |\bar{0}\rangle |0\rangle \tag{3}$$

in time $O(\operatorname{poly} \log n)$, where $s_i \triangleq \operatorname{sign}(u_i)$. On the other hand, using query access to $\mathbf{A}$ create the mapping $|i\rangle|s_i\rangle|\bar{0}\rangle \mapsto |i\rangle|s_i\rangle|s_i A_{ij}\rangle$ costing time $O(\operatorname{poly} \log(nd))$. Using this operator on the first three registers in Equation (3), we obtain

$$\sum_{i=1}^{n} \sqrt{\frac{|u_i|}{\|\mathbf{u}\|_1}} |i\rangle |s_i\rangle |s_i A_{ij}\rangle |0\rangle. \tag{4}$$

Now, define the positive-controlled rotation for each $a \in \mathbb{R}$ such that

$$U_{\mathrm{CR}^+} : |a\rangle|0\rangle \mapsto \begin{cases} |a\rangle(\sqrt{a}|1\rangle + \sqrt{1-a}|0\rangle) & \text{if } a \in (0,1], \\ |a\rangle|0\rangle & \text{otherwise.} \end{cases}$$

Applying U_{CR^+} on the last two registers in Equation (4), we obtain

$$\begin{aligned}
&\sum_{\substack{i\in[n] \\ s_i A_{ij} > 0}} \sqrt{\frac{|u_i|}{\|\mathbf{u}\|_1}} |i\rangle|s_i\rangle|s_i A_{ij}\rangle \left(\sqrt{\frac{s_i A_{ij}}{\|\mathbf{A}_j\|_\infty}} |1\rangle + \sqrt{1 - \frac{s_i A_{ij}}{\|\mathbf{A}_j\|_\infty}} |0\rangle \right) + \sum_{\substack{i\in[n] \\ s_i A_{ij} \leqslant 0}} \sqrt{\frac{|u_i|}{\|\mathbf{u}\|_1}} |i\rangle|s_i\rangle|s_i A_{ij}\rangle|0\rangle \\
&= \sum_{\substack{i\in[n] \\ s_i A_{ij} > 0}} \sqrt{\frac{A_{ij} u_i}{\|\mathbf{A}_j\|_\infty \|\mathbf{u}\|_1}} |i\rangle|s_i\rangle|s_i A_{ij}\rangle|1\rangle \\
&\qquad + \left(\sum_{\substack{i\in[n] \\ s_i A_{ij} > 0}} \sqrt{\frac{|u_i|}{\|\mathbf{u}\|_1} - \frac{A_{ij} u_i}{\|\mathbf{A}_j\|_\infty \|\mathbf{u}\|_1}} |i\rangle|s_i\rangle|s_i A_{ij}\rangle + \sum_{\substack{i\in[n] \\ s_i A_{ij} \leqslant 0}} \sqrt{\frac{|u_i|}{\|\mathbf{u}\|_1}} |i\rangle|s_i\rangle|s_i A_{ij}\rangle \right) |0\rangle \\
&= \sqrt{a_+}|\phi_1\rangle|1\rangle + \sqrt{1-a_+}|\phi_0\rangle|0\rangle,
\end{aligned}$$

where

$$a_+ = \sum_{\substack{i\in[n] \\ s_i A_{ij} > 0}} \frac{A_{ij} u_i}{\|\mathbf{A}_j\|_\infty \|\mathbf{u}\|_1} \quad \text{and} \quad |\phi_1\rangle = \sum_{\substack{i\in[n] \\ s_i A_{ij} > 0}} \sqrt{\frac{A_{ij} u_i}{\sum_{k\in[n], s_k A_{kj} > 0} A_{kj} u_k}} |i\rangle|s_i\rangle|s_i A_{ij}\rangle.$$

Here, $|\phi_1\rangle$ and $|\phi_0\rangle$ are unit vectors. We now use Fact 12 to get an estimate $\widetilde{a}_+$ such that

$$|\widetilde{a}_+ - a_+| \leqslant \frac{\epsilon}{2}\sqrt{a_+} \implies \left| \|\mathbf{A}_j\|_\infty \|\mathbf{u}\|_1 \widetilde{a}_+ - \sum_{\substack{i\in[n] \\ s_i A_{ij} > 0}} A_{ij} u_i \right| \leqslant \frac{\epsilon}{2} \sqrt{\|\mathbf{A}_j\|_\infty \|\mathbf{u}\|_1 \sum_{\substack{i\in[n] \\ s_i A_{ij} > 0}} A_{ij} u_i} \leqslant \frac{\epsilon}{2} \|\mathbf{A}_j\|_\infty \|\mathbf{u}\|_1$$

in time $O(\epsilon^{-1} \log(1/\delta) \operatorname{poly} \log(nd))$ and probability at least $1 - \delta/2$. Notice that we know $\|\mathbf{u}\|_1$ since it is stored in a KP-tree. Similarly, we estimate

$$a_- = - \sum_{\substack{i\in[n] \\ s_i A_{ij} < 0}} \frac{A_{ij} u_i}{\|\mathbf{A}_j\|_\infty \|\mathbf{u}\|_1}$$

with additive error $\frac{\epsilon}{2}\|\mathbf{A}_j\|_\infty\|\mathbf{u}\|_1$. Thus $\mathbf{A}_j^\top\mathbf{u} = \sum_{i=1}^n A_{ij}u_i = \|\mathbf{A}_j\|_\infty\|\mathbf{u}\|_1(a_+ - a_-)$ is estimated with additive error $\epsilon\|\mathbf{A}_j\|_\infty\|\mathbf{u}\|_1$ and success probability at least $1-\delta$ in time $O(\epsilon^{-1}\log(1/\delta)\operatorname{poly}\log(nd))$.

Swapping the roles of $\mathbf{A}_j$ and $\mathbf{u}$ and repeating the above steps leads to an estimate of $\mathbf{A}_j^\top\mathbf{u}$ with additive error $\epsilon\|\mathbf{A}_j\|_1\|\mathbf{u}\|_\infty$ in time $O(\epsilon^{-1}\log(1/\delta)\operatorname{poly}\log(nd))$. □

It is possible to prove a classical result analogous to the one above. More specifically, if we have sampling access to $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $\mathbf{u} \in \mathbb{R}^n$, then there is a classical algorithm that allows to approximate $\mathbf{A}_j^\top\mathbf{u}$ up to an additive error.

Lemma 14. *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $\mathbf{u} \in \mathbb{R}^n$. Assume sampling access to $\mathbf{u}$ and to the columns of $\mathbf{A}$. Let $\delta \in (0,1)$ and $\epsilon > 0$. There is a classical algorithm that, for any $j \in [d]$, outputs r_j such that*

$$|r_j - \mathbf{A}_j^\top\mathbf{u}| \leqslant \epsilon \min(\|\mathbf{A}_j\|_\infty\|\mathbf{u}\|_1, \|\mathbf{A}_j\|_1\|\mathbf{u}\|_\infty)$$

with probability at least $1-\delta$ and in time $O(\epsilon^{-2}\log(1/\delta)\operatorname{poly}\log(nd))$.

Proof. Fix $j \in [d]$. Let z be the random variable defined by

$$\mathbb{P}\left[z = \|\mathbf{A}_j\|_1 u_i \operatorname{sign}(A_{ij})\right] = \frac{|A_{ij}|}{\|\mathbf{A}_j\|_1} \quad \text{for } i \in [n].$$

Then

$$\mathbb{E}[z] = \sum_{i=1}^n \frac{|A_{ij}|}{\|\mathbf{A}_j\|_1}\|\mathbf{A}_j\|_1 u_i \operatorname{sign}(A_{ij}) = \mathbf{A}_j^\top\mathbf{u}.$$

Sample $q = \lceil 2\epsilon^{-2}\ln(2/\delta)\rceil$ indices $\{i_1,\dots,i_q\} \subseteq [n]$ from the distribution $\mathcal{D}_{\mathbf{A}_j}$ by using sampling access to $\mathbf{A}$ and set $z_k = \|\mathbf{A}_j\|_1 u_{i_k}\operatorname{sign}(A_{i_k j})$, $k \in [q]$. The algorithm outputs $\widehat{z} = \frac{1}{q}\sum_{k=1}^q z_k$ as an estimate for $\mathbf{A}_j^\top\mathbf{u}$. Since $z_1,\dots,z_q$ are i.i.d. copies of z, a Hoeffding's bound (Fact 5) gives

$$\mathbb{P}[|\widehat{z} - \mathbf{A}_j^\top\mathbf{u}| \geqslant \epsilon\|\mathbf{A}_j\|_1\|\mathbf{u}\|_\infty] \leqslant 2e^{-\epsilon^2 q/2} = \delta,$$

so the classical algorithm approximates $\mathbf{A}_j^\top\mathbf{u}$ with additive error $\epsilon\|\mathbf{A}_j\|_1\|\mathbf{u}\|_\infty$ and success probability at least $1-\delta$. The final runtime is the number of samples q times the complexity $O(\operatorname{poly}\log(nd))$ of sampling each index. Finally, it is possible to repeat the same procedure but swapping the roles of $\mathbf{A}_j$ and $\mathbf{u}$, so that z is now defined as $\mathbb{P}\left[z = \|\mathbf{u}\|_1 A_{ij}\operatorname{sign}(u_i)\right] = |u_i|/\|\mathbf{u}\|_1$, $i \in [n]$. □

3 Lasso path and the LARS algorithm

3.1 Lasso solution

In this section, we review several known properties of the solution to the Lasso regression problem with inputs $(\mathbf{X},\mathbf{y}) \in \mathbb{R}^{n\times d}\times\mathbb{R}^n$ defined as

$$\hat{\boldsymbol{\beta}} \in \operatorname*{arg\,min}_{\boldsymbol{\beta}\in\mathbb{R}^d} \mathcal{L}^{(\lambda)}_{\mathbf{X},\mathbf{y}}(\boldsymbol{\beta}) \quad \text{where} \quad \mathcal{L}^{(\lambda)}_{\mathbf{X},\mathbf{y}}(\boldsymbol{\beta}) \triangleq \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \quad \text{for } \lambda > 0. \tag{5}$$

Here $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the design matrix and $\mathbf{y} \in \mathbb{R}^n$ is a vector of observations. When the inputs are clear from context, we shall simply write $\mathcal{L}^{(\lambda)}(\boldsymbol{\beta})$. We will cover optimality conditions for the Lasso problem, the general form for its solution, and continuity and uniqueness conditions. Most of these results can be found in [RZ07, MY12, Tib13, SCL+18]. We start with obtaining the optimality conditions for the Lasso problem.

Fact 15 ([Tib13, Lemma 1])**.** *Every Lasso solution $\hat{\boldsymbol{\beta}} \in \mathbb{R}^d$ gives the same fitted value $\mathbf{X}\hat{\boldsymbol{\beta}}$.*

Fact 16. *A vector $\hat{\boldsymbol{\beta}} \in \mathbb{R}^d$ is a solution to the Lasso problem if and only if*

$$\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \lambda \mathbf{s} \quad \text{where } s_i \in \begin{cases} \{\operatorname{sign}(\hat{\beta}_i)\} & \text{if } \hat{\beta}_i \neq 0, \\ [-1, 1] & \text{if } \hat{\beta}_i = 0, \end{cases} \quad \text{for } i \in [d]. \tag{6}$$

Let $\mathcal{A} \triangleq \{i \in [d] : |\mathbf{X}_i^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})| = \lambda\}$, $\mathcal{I} \triangleq [d] \backslash \mathcal{A}$, and $\boldsymbol{\eta} \triangleq \operatorname{sign}(\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})) \in \{-1, 0, 1\}^d$. Any Lasso solution $\hat{\boldsymbol{\beta}}$ is of the form

$$\hat{\boldsymbol{\beta}}_{\mathcal{I}} = \mathbf{0} \quad \text{and} \quad \hat{\boldsymbol{\beta}}_{\mathcal{A}} = \mathbf{X}_{\mathcal{A}}^+(\mathbf{y} - \lambda(\mathbf{X}_{\mathcal{A}}^+)^\top \boldsymbol{\eta}_{\mathcal{A}}) + \mathbf{b}, \tag{7}$$

where

$$\mathbf{b} \in \operatorname{null}(\mathbf{X}_{\mathcal{A}}) \quad \text{and} \quad \eta_i\big([\mathbf{X}_{\mathcal{A}}^+(\mathbf{y} - \lambda(\mathbf{X}_{\mathcal{A}}^+)^\top \boldsymbol{\eta}_{\mathcal{A}})]_i + b_i\big) \geqslant 0 \text{ for } i \in \mathcal{A}. \tag{8}$$

Proof. Equation (6) can be obtained by considering the Karush–Kuhn–Tucker conditions, or sub-gradient optimality conditions. More specifically, a vector $\hat{\boldsymbol{\beta}} \in \mathbb{R}^d$ is a solution to the Lasso problem if and only if (see, e.g., [BL06, Proposition 3.1.5])

$$\mathbf{0} \in \{-\nabla \mathcal{L}^{(\lambda)}(\hat{\boldsymbol{\beta}}) = \mathbf{X}^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) - \lambda \mathbf{s} : \mathbf{s} \in \partial \|\hat{\boldsymbol{\beta}}\|_1\},$$

where $\partial\|\hat{\boldsymbol{\beta}}\|_1$ denotes the sub-gradient of the ℓ_1-norm at $\hat{\boldsymbol{\beta}}$. Equation (6) then follows from the fact that the sub-gradients $\mathbf{s} \in \mathbb{R}^d$ of $\|\hat{\boldsymbol{\beta}}\|_1$ are vectors such that $s_i = \operatorname{sign}(\hat{\beta}_i)$ if $\hat{\beta}_i \neq 0$ and $|s_i| \leqslant 1$ otherwise.

Moving on, first note that $\boldsymbol{\eta}$ is a sub-gradient and that the uniqueness of $\mathbf{X}\hat{\boldsymbol{\beta}}$ (Fact 15) implies the uniqueness of $\mathcal{A}$ and $\boldsymbol{\eta}$. By the definition of the sub-gradients $\mathbf{s}$ in Equation (6), we know that $\hat{\boldsymbol{\beta}}_{\mathcal{I}} = \mathbf{0}$, and therefore the block $\mathcal{A}$ of Equation (6) can be written as

$$\mathbf{X}_{\mathcal{A}}^\top(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\hat{\boldsymbol{\beta}}_{\mathcal{A}}) = \lambda \boldsymbol{\eta}_{\mathcal{A}}.$$

This means that $\boldsymbol{\eta}_{\mathcal{A}} \in \operatorname{row}(\mathbf{X}_{\mathcal{A}})$, and so $\boldsymbol{\eta}_{\mathcal{A}} = \mathbf{X}_{\mathcal{A}}^\top(\mathbf{X}_{\mathcal{A}}^\top)^+ \boldsymbol{\eta}_{\mathcal{A}}$. Using this fact and rearranging the above equation, we obtain

$$\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}} \hat{\boldsymbol{\beta}}_{\mathcal{A}} = \mathbf{X}_{\mathcal{A}}^\top(\mathbf{y} - \lambda(\mathbf{X}_{\mathcal{A}}^\top)^+ \boldsymbol{\eta}_{\mathcal{A}}).$$

Therefore, any solution $\hat{\boldsymbol{\beta}}_{\mathcal{A}}$ to the above equation is of the form

$$\hat{\boldsymbol{\beta}}_{\mathcal{A}} = \mathbf{X}_{\mathcal{A}}^+(\mathbf{y} - \lambda(\mathbf{X}_{\mathcal{A}}^\top)^+ \boldsymbol{\eta}_{\mathcal{A}}) + \mathbf{b}, \qquad \mathbf{b} \in \operatorname{null}(\mathbf{X}_{\mathcal{A}}),$$

where we used the identity $\mathbf{A}^+(\mathbf{A}^\top)^+\mathbf{A}^\top = \mathbf{A}^+$. Any $\mathbf{b} \in \operatorname{null}(\mathbf{X}_{\mathcal{A}})$ produces a valid Lasso solution $\hat{\boldsymbol{\beta}}_{\mathcal{A}}$ as long as $\hat{\boldsymbol{\beta}}_{\mathcal{A}}$ has the correct signs given by $\boldsymbol{\eta}_{\mathcal{A}}$, i.e., $\operatorname{sign}(\hat{\beta}_i) = \eta_i$ for $i \in \mathcal{A}$. This restriction can be written as $\hat{\beta}_i \eta_i \geqslant 0$, which is Equation (8). This completes the proof. □

We shall abuse the definition of the Karush-Kuhn-Tucker (KKT) conditions and refer to Equation (6) as the KKT conditions themselves for the Lasso problem. The set

$$\mathcal{A} \triangleq \{i \in [d] : |\mathbf{X}_i^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})| = \lambda\}$$

and the sub-gradient vector

$$\boldsymbol{\eta} \triangleq \operatorname{sign}(\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})) \in \{-1, 0, 1\}^d$$

are normally referred to as *active* (or *equicorrelation*) *set* and *equicorrelation signs*, respectively. The set $\mathcal{I} \triangleq [d] \backslash \mathcal{A}$ is called *inactive set*.

Fact 17 ([Tib13, Lemma 9])**.** *For any* $\mathbf{y} \in \mathbb{R}^n$*,* $\mathbf{X} \in \mathbb{R}^{n \times d}$*, and* $\lambda > 0$*, the Lasso solution*

$$\hat{\boldsymbol{\beta}}_{\mathcal{I}} = \mathbf{0} \quad \textit{and} \quad \hat{\boldsymbol{\beta}}_{\mathcal{A}} = \mathbf{X}_{\mathcal{A}}^{+}(\mathbf{y} - \lambda(\mathbf{X}_{\mathcal{A}}^{+})^{\top}\boldsymbol{\eta}_{\mathcal{A}})$$

has the minimum ℓ_2*-norm over all Lasso solutions.*

Proof. By Fact 16, any Lasso solution has ℓ_2-norm

$$\|\hat{\boldsymbol{\beta}}\|_2^2 = \|\mathbf{X}_{\mathcal{A}}^{+}(\mathbf{y} - \lambda(\mathbf{X}_{\mathcal{A}}^{+})^{\top}\boldsymbol{\eta}_{\mathcal{A}})\|^2 + \|\mathbf{b}\|^2,$$

since $\mathbf{b} \in \operatorname{null}(\mathbf{X}_{\mathcal{A}})$. Hence the ℓ_2-norm is minimised when $\mathbf{b} = \mathbf{0}$. □

It follows from Fact 16 that the Lasso solution is unique if $\operatorname{null}(\mathbf{X}_{\mathcal{A}}) = \{\mathbf{0}\}$ and is given by

$$\hat{\boldsymbol{\beta}}_{\mathcal{I}} = \mathbf{0} \quad \text{and} \quad \hat{\boldsymbol{\beta}}_{\mathcal{A}} = (\mathbf{X}_{\mathcal{A}}^{\top}\mathbf{X}_{\mathcal{A}})^{-1}(\mathbf{X}_{\mathcal{A}}^{\top}\mathbf{y} - \lambda\boldsymbol{\eta}_{\mathcal{A}}). \tag{9}$$

It is known [Tib13, Lemma 14] that there always is a Lasso solution such that $|\mathcal{A}| \leqslant \min\{n, d\}$. If the Lasso solution is unique, then the active set has size at most $\min\{n, d\}$. The sufficient condition $\operatorname{null}(\mathbf{X}_{\mathcal{A}}) = \{\mathbf{0}\}$ for uniqueness has appeared several times in the literature [OPT00a, Fuc05, Wai09, CP09, Tib13]. Since the active set $\mathcal{A}$ depends on the Lasso solution at $\mathbf{y}, \mathbf{X}, \lambda$, the condition that $\mathbf{X}_{\mathcal{A}}$ has full rank is somewhat circular. Because of this, more natural conditions are assumed which imply $\operatorname{null}(\mathbf{X}_{\mathcal{A}}) = \{\mathbf{0}\}$, e.g., the general position property [Ros04, Don06, Dos12, Tib13] and the matrix $\mathbf{X}$ being drawn from a continuous probability distribution [Tib13]. We say that a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ has columns in *general position* if no k-dimensional subspace in $\mathbb{R}^n$, for $k < \min\{n, d\}$, contains more than $k + 1$ elements from the set $\{\pm\mathbf{X}_1, \ldots, \pm\mathbf{X}_d\}$, excluding antipodal pairs. It is known [Ros04, Don06, Dos12, Tib13] that this is enough to guarantee the uniqueness of the Lasso solution.

Fact 18 ([Tib13, Lemma 3])**.** *If the columns of* $\mathbf{X} \in \mathbb{R}^{n \times d}$ *are in general position, then the Lasso solution is unique for any* $\mathbf{y} \in \mathbb{R}^n$ *and* $\lambda > 0$ *and is given by* Equation (9)*.*

Another sufficient condition was given by Tibshirani [Tib13], who proved the following.

Fact 19 ([Tib13, Lemma 4])**.** *If the entries of* $\mathbf{X} \in \mathbb{R}^{n \times d}$ *are drawn from a continuous probability distribution on* $\mathbb{R}^{nd}$*, then the Lasso solution is unique for any* $\mathbf{y} \in \mathbb{R}^n$ *and* $\lambda > 0$ *and is given by* Equation (9) *with probability* 1*.*

It is well known that, when $\operatorname{null}(\mathbf{X}_{\mathcal{A}}) = \{\mathbf{0}\}$, the KKT conditions from Fact 16 imply that the regularisation path $\mathcal{P} \triangleq \{\hat{\boldsymbol{\beta}}(\lambda) \in \mathbb{R}^d : \lambda > 0\}$ comprising all the solutions for $\lambda > 0$ is continuous piecewise linear, which was first proven by Efron, Hastie, Johnstone, and Tibshirani [EHJT04], cf. Equation (2). This fact was later extended to the case when $\operatorname{rank}(\mathbf{X}_{\mathcal{A}}) < |\mathcal{A}|$ by Tibshirani [Tib13], who proved that the path of solutions with $\mathbf{b} = \mathbf{0}$ in Equation (7) is continuous and piecewise linear.

Fact 20 ([Tib13, Appendix A])**.** *The regularisation path* $\mathcal{P} = \{\hat{\boldsymbol{\beta}}(\lambda) \in \mathbb{R}^d : \lambda > 0\}$ *formed by the Lasso solution*

$$\hat{\boldsymbol{\beta}}_{\mathcal{I}}(\lambda) = \mathbf{0} \quad \textit{and} \quad \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda) = \mathbf{X}_{\mathcal{A}}^{+}(\mathbf{y} - \lambda(\mathbf{X}_{\mathcal{A}}^{+})^{\top}\boldsymbol{\eta}_{\mathcal{A}}),$$

with $\mathcal{A} = \{i \in [d] : |\mathbf{X}_i^{\top}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda))| = \lambda\}$ *and* $\mathcal{I} = [d] \backslash \mathcal{A}$*, is continuous piecewise linear.*

We call *kink* a point where the direction $\partial\hat{\boldsymbol{\beta}}(\lambda)/\partial\lambda$ changes. The path $\{\hat{\boldsymbol{\beta}}(\lambda) : \lambda_{t+1} < \lambda \leqslant \lambda_t\}$ between two consecutive kinks λ_{t+1} and λ_t thus defines a linear segment. An important consequence of the above result is that any continuous piecewise linear regularisation path has at most 3^d linear segments, since two different linear segments must have different equicorrelation signs $\boldsymbol{\eta} \in \{-1, 0, 1\}^d$. This result was improved slightly to $(3^d + 1)/2$ iterations by Mairal and Yu [MY12], who also exhibited a worst-case Lasso problem with exactly $(3^d + 1)/2$ number of linear segments.

3.2 The LARS algorithm

The Lasso solution path $\{\hat{\boldsymbol{\beta}}(\lambda) : \lambda > 0\}$ can be computed by the Least Angle Regression (LARS) algorithm[2] which was proposed and named by Efron *et al.* [EHJT04], although similar ideas have already appeared in the work of Osborne *et al.* [OPT00a, OPT00b]. Since its introduction, the LARS algorithm have been improved and generalised [RZ07, MY12, Tib13, SCL+18]. In compressed sensing literature, this algorithm is better known as the Homotopy method [FR13].

Starting at $\lambda = \infty$, where the Lasso solution is trivially $\mathbf{0} \in \mathbb{R}^d$, the LARS algorithm iteratively computes all the kinks in the Lasso solution path by decreasing the parameter λ and checking for points where the path's linear trajectory changes. This is done by making sure that the KKT conditions from Fact 16 are always satisfied. When a kink is reached, a coordinate from the solution $\hat{\boldsymbol{\beta}}(\lambda)$ will go from non-zero (being in the active set $\mathcal{A}$) to zero (join the inactive set $\mathcal{I}$), or vice-versa, i.e., from zero (leave $\mathcal{I}$) to non-zero (into $\mathcal{A}$). We shall now describe the LARS algorithm in more details, following [Tib13]. Algorithm 1 summarises the LARS algorithm.

The LARS algorithm starts, without loss of generality, at the first kink $\lambda_0 = \|\mathbf{X}^\top \mathbf{y}\|_\infty$, since $\hat{\boldsymbol{\beta}}(\lambda) = \mathbf{0}$ for $\lambda \geqslant \lambda_0$ (plug $\hat{\boldsymbol{\beta}}(\lambda_0) = \mathbf{0}$ in Equation (6)). The corresponding active and inactive sets are $\mathcal{A} = \operatorname{argmax}_{j \in [d]} |\mathbf{X}_j^\top \mathbf{y}|$ and $\mathcal{I} = [d] \backslash \mathcal{A}$, respectively, while the equicorrelation sign is $\boldsymbol{\eta}_\mathcal{A} = \operatorname{sign}(\mathbf{X}_\mathcal{A}^\top \mathbf{y})$. Since every iteration after the initial setup is identical, assume by induction that the LARS algorithm has computed the first $t+1$ kinks $(\lambda_0, \hat{\boldsymbol{\beta}}(\lambda_0)), \ldots, (\lambda_t, \hat{\boldsymbol{\beta}}(\lambda_t))$. At the beginning of the t-th iteration, the regularisation parameter is $\lambda = \lambda_t$, the Lasso solution is $\hat{\boldsymbol{\beta}}(\lambda_t)$, and the active and inactive sets are $\mathcal{A}$ and $\mathcal{I}$. From the previous solution $\hat{\boldsymbol{\beta}}(\lambda_t)$ we obtain the equicorrelation signs

$$\boldsymbol{\eta}_\mathcal{A} = \operatorname{sign}\left(\mathbf{X}_\mathcal{A}^\top (\mathbf{y} - \mathbf{X}_\mathcal{A} \hat{\boldsymbol{\beta}}_\mathcal{A}(\lambda_t))\right).$$

According to Equation (7), for $\lambda \leqslant \lambda_t$ sufficiently close to λ_t, the Lasso solution is[3]

$$\hat{\boldsymbol{\beta}}_\mathcal{I}(\lambda) = \mathbf{0} \quad \text{and} \quad \hat{\boldsymbol{\beta}}_\mathcal{A}(\lambda) = \mathbf{X}_\mathcal{A}^+ (\mathbf{y} - \lambda (\mathbf{X}_\mathcal{A}^+)^\top \boldsymbol{\eta}_\mathcal{A}) = \boldsymbol{\mu} - \lambda \boldsymbol{\theta}, \tag{10}$$

where $\boldsymbol{\mu} = \mathbf{X}_\mathcal{A}^+ \mathbf{y}$ and $\boldsymbol{\theta} = (\mathbf{X}_\mathcal{A}^\top \mathbf{X}_\mathcal{A})^+ \boldsymbol{\eta}_\mathcal{A}$. We now decrease λ while keeping Equation (10). As λ decreases, two important checks must be made. First, we check when (i.e., we compute the next value of λ at which) a variable i_{t+1}^{join} in $\mathcal{I}$ should join the active set $\mathcal{A}$ because it has attain the maximum correlation $|\mathbf{X}_{i_{t+1}^{\text{join}}}^\top (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})| = \lambda$. We call this event the next *joining time* $\lambda_{t+1}^{\text{join}}$. Second, we check when a variable i_{t+1}^{cross} in the active set $\mathcal{A}$ should leave $\mathcal{A}$ and join $\mathcal{I}$ because $\hat{\beta}_{i_{t+1}^{\text{cross}}}$ crossed through zero. We call this event the next *crossing time* $\lambda_{t+1}^{\text{cross}}$.

For the first check, for each $i \in \mathcal{I}$, consider the equation

$$\mathbf{X}_i^\top (\mathbf{y} - \mathbf{X}_\mathcal{A} \hat{\boldsymbol{\beta}}_\mathcal{A}(\lambda)) = \pm\lambda \implies \mathbf{X}_i^\top (\mathbf{y} - \mathbf{X}_\mathcal{A}(\boldsymbol{\mu} - \lambda\boldsymbol{\theta})) = \pm\lambda.$$

The solution to the above equation for λ, interpreted as the joining time of the i-th variable, is

$$\Lambda_i^{\text{join}}(\mathcal{A}) \triangleq \frac{\mathbf{X}_i^\top (\mathbf{y} - \mathbf{X}_\mathcal{A}\boldsymbol{\mu})}{\pm 1 - \mathbf{X}_i^\top \mathbf{X}_\mathcal{A} \boldsymbol{\theta}} = \frac{\mathbf{X}_i^\top (\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}}{\pm 1 - \mathbf{X}_i^\top (\mathbf{X}_\mathcal{A}^\top)^+ \boldsymbol{\eta}_\mathcal{A}}, \tag{11}$$

where we used that $\mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+ (\mathbf{X}_\mathcal{A}^\top)^+ = (\mathbf{X}_\mathcal{A}^\top)^+$. Only one of the possibilities $+1$ or -1 will lead to a value in the necessary interval $[0, \lambda_t]$, and the following expressions always consider that solution.

[2] By LARS algorithm we mean the version of the algorithm that computes the Lasso path and not the version that performs a kind of forward variable selection.

[3] From now on, we shall assume $\mathbf{b} = \mathbf{0}$ in order to guarantee the continuity of the Lasso path.

Algorithm 1: Classical LARS algorithm for the pathwise Lasso

Input: Vector $\mathbf{y} \in \mathbb{R}^n$ and matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$

Output: Exact regularisation path $\mathcal{P}$

Initialise $\mathcal{A} = \operatorname{argmax}_{j \in [d]} |\mathbf{X}_j^\top \mathbf{y}|$, $\mathcal{I} = [d] \backslash \mathcal{A}$, $\lambda_0 = \|\mathbf{X}^\top \mathbf{y}\|_\infty$, $\hat{\boldsymbol{\beta}}(\lambda_0) = \mathbf{0}$, $t = 0$

while $\mathcal{I} \neq \varnothing$ **do**

  $\boldsymbol{\eta}_\mathcal{A} \leftarrow \operatorname{sign}\left(\mathbf{X}_\mathcal{A}^\top (\mathbf{y} - \mathbf{X}_\mathcal{A} \hat{\boldsymbol{\beta}}_\mathcal{A}(\lambda_t))\right)$

  $\boldsymbol{\mu} \leftarrow \mathbf{X}_\mathcal{A}^+ \mathbf{y}$ // $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{A}|}$

  $\boldsymbol{\theta} \leftarrow (\mathbf{X}_\mathcal{A}^\top \mathbf{X}_\mathcal{A})^+ \boldsymbol{\eta}_\mathcal{A}$ // $\boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{A}|}$

  $\Lambda_i^{\text{cross}} \leftarrow \frac{\mu_i}{\theta_i} \cdot \mathbf{1}\left[\frac{\mu_i}{\theta_i} \leqslant \lambda_t\right] \quad \forall i \in \mathcal{A}$

  $\Lambda_i^{\text{join}} \leftarrow \frac{\mathbf{X}_i^\top (\mathbf{y} - \mathbf{X}_\mathcal{A} \boldsymbol{\mu})}{\pm 1 - \mathbf{X}_i^\top \mathbf{X}_\mathcal{A} \boldsymbol{\theta}} \quad \forall i \in \mathcal{I}$

  $i_{t+1}^{\text{cross}} \leftarrow \operatorname{argmax}_{i \in \mathcal{A}} \{\Lambda_i^{\text{cross}}\}$ and $\lambda_{t+1}^{\text{cross}} \leftarrow \max_{i \in \mathcal{A}} \{\Lambda_i^{\text{cross}}\}$

  $i_{t+1}^{\text{join}} \leftarrow \operatorname{argmax}_{i \in \mathcal{I}} \{\Lambda_i^{\text{join}}\}$ and $\lambda_{t+1}^{\text{join}} \leftarrow \max_{i \in \mathcal{I}} \{\Lambda_i^{\text{join}}\}$

  $\lambda_{t+1} \leftarrow \max\{\lambda_{t+1}^{\text{join}}, \lambda_{t+1}^{\text{cross}}\}$

  $\hat{\boldsymbol{\beta}}_\mathcal{A}(\lambda_{t+1}) \leftarrow \boldsymbol{\mu} - \lambda_{t+1} \boldsymbol{\theta}$

  if $\lambda_{t+1} = \lambda_{t+1}^{\text{cross}}$ **then**

    Move i_{t+1}^{cross} from $\mathcal{A}$ to $\mathcal{I}$

  else

    Move i_{t+1}^{join} from $\mathcal{I}$ to $\mathcal{A}$

  $t \leftarrow t + 1$

return coefficients $[(\lambda_0, \hat{\boldsymbol{\beta}}(\lambda_0)), (\lambda_1, \hat{\boldsymbol{\beta}}(\lambda_1)), \dots]$

Therefore, the next joining time $\lambda_{t+1}^{\text{join}}$ and coordinate i_{t+1}^{join} are

$$\lambda_{t+1}^{\text{join}} = \max_{i \in \mathcal{I}} \{\Lambda_i^{\text{join}}(\mathcal{A})\} \quad \text{and} \quad i_{t+1}^{\text{join}} = \operatorname*{argmax}_{i \in \mathcal{I}} \{\Lambda_i^{\text{join}}(\mathcal{A})\}.$$

For the second check, for each $i \in \mathcal{A}$, we solve the equation

$$\hat{\beta}_i(\lambda) = 0 \implies \mu_i - \lambda \theta_i = 0.$$

The crossing time of the i-th variable, which is the solution to the above equation restricted to values $\lambda \leqslant \lambda_t$, is thus defined as

$$\Lambda_i^{\text{cross}}(\mathcal{A}) \triangleq \frac{\mu_i}{\theta_i} \cdot \mathbf{1}\left[\frac{\mu_i}{\theta_i} \leqslant \lambda_t\right] = \frac{[\mathbf{X}_\mathcal{A}^+ \mathbf{y}]_i}{[(\mathbf{X}_\mathcal{A}^\top \mathbf{X}_\mathcal{A})^+ \boldsymbol{\eta}_\mathcal{A}]_i} \cdot \mathbf{1}\left[\frac{[\mathbf{X}_\mathcal{A}^+ \mathbf{y}]_i}{[(\mathbf{X}_\mathcal{A}^\top \mathbf{X}_\mathcal{A})^+ \boldsymbol{\eta}_\mathcal{A}]_i} \leqslant \lambda_t\right].$$

Therefore, the next crossing time $\lambda_{t+1}^{\text{join}}$ and coordinate i_{t+1}^{join} are

$$\lambda_{t+1}^{\text{cross}} = \max_{i \in \mathcal{A}} \{\Lambda_i^{\text{cross}}(\mathcal{A})\} \quad \text{and} \quad i_{t+1}^{\text{cross}} = \operatorname*{argmax}_{i \in \mathcal{A}} \{\Lambda_i^{\text{cross}}(\mathcal{A})\}.$$

Finally, the next kink of the regularisation path is $(\lambda_{t+1}, \hat{\boldsymbol{\beta}}(\lambda_{t+1}))$ where $\lambda_{t+1} = \max\{\lambda_{t+1}^{\text{join}}, \lambda_{t+1}^{\text{cross}}\}$ and $\hat{\boldsymbol{\beta}}_\mathcal{I}(\lambda_{t+1}) = \mathbf{0}$ and $\hat{\boldsymbol{\beta}}_\mathcal{A}(\lambda_{t+1}) = \boldsymbol{\mu} - \lambda_{t+1} \boldsymbol{\theta}$. If $\lambda_{t+1}^{\text{join}} > \lambda_{t+1}^{\text{cross}}$, then the coordinate i_{t+1}^{join} should join the active set $\mathcal{A}$. Otherwise, the coordinate i_{t+1}^{cross} should leave the active set $\mathcal{A}$. This ends the t-th iteration of the LARS algorithm.

The time complexity of each iteration in the LARS algorithm is given by the following result.

Fact 21. *The time complexity per iteration of the* LARS *algorithm is* $O(nd + |\mathcal{A}|^2)$*, where* $\mathcal{A}$ *is the active set of the corresponding iteration.*

Proof. Let us start with the computation of $\mathbf{X}_{\mathcal{A}}^{+}$. Simply computing $\mathbf{X}_{\mathcal{A}}^{+}$ would require $O(n|\mathcal{A}|^2 + |\mathcal{A}|^3)$ time. However, the time complexity can be reduced to $O(n|\mathcal{A}| + |\mathcal{A}|^2)$ by using the Sherman–Morrison formula applied to the Moore-Penrose inverse. More specifically, suppose we have previously computed $\mathbf{X}_{\mathcal{A}}^{+}$ and the index i_{t+1}^{join} is then added to the active set $\mathcal{A}$ to obtain the new active set $\mathcal{A}' = \mathcal{A} \cup \{i_{t+1}^{\text{join}}\}$. Without lost of generality, write $\mathcal{A} = \{j_1, \dots, j_k\}$ and $\mathcal{A}' = \{j_1, \dots, j_k, i_{t+1}^{\text{join}}\}$, where $k = |\mathcal{A}|$. Let $\mathbf{A} \in \mathbb{R}^{n \times (|\mathcal{A}|+1)}$ be the matrix $\mathbf{X}_{\mathcal{A}}$ augmented with a zero column at the position corresponding to i_{t+1}^{join}, i.e., $\mathbf{A}_\ell = \mathbf{X}_{j_\ell}$ if $\ell \in [k]$, and $\mathbf{A}_\ell = \mathbf{0}$ if $\ell = k+1$. The augmented matrix is written using a rank-1 update as

$$\mathbf{X}_{\mathcal{A}'} = \mathbf{A} + \mathbf{X}_{i_{t+1}^{\text{join}}} \mathbf{e}_{k+1}^{\top},$$

where $\mathbf{e}_\ell \in \{0,1\}^{|\mathcal{A}|+1}$ is the column vector with 1 in position ℓ and 0 elsewhere. Therefore, by using the Sherman–Morrison formula applied to the Moore-Penrose inverse (see [Mey73]), the inverse $\mathbf{X}_{\mathcal{A}'}^{+}$ can be computed from the matrix $\mathbf{X}_{\mathcal{A}}^{+}$ and the vectors $\mathbf{X}_{i_{t+1}^{\text{join}}}$ and $\mathbf{e}_{k+1}$ using simple matrix and vector multiplication, which requires $O(n|\mathcal{A}| + |\mathcal{A}|^2)$ time. Regarding the other quantities,

- Computing $\boldsymbol{\eta}_{\mathcal{A}}$, $\boldsymbol{\mu}$, and $\boldsymbol{\theta}$ requires $O(n|\mathcal{A}| + |\mathcal{A}|^2)$ time (the updating argument can be used as well for $(\mathbf{X}_{\mathcal{A}}^{\top}\mathbf{X}_{\mathcal{A}})^{+}$);
- Finding i_{t+1}^{cross} and $\lambda_{t+1}^{\text{cross}}$ requires $O(|\mathcal{A}|)$ time;
- Finding i_{t+1}^{join} and $\lambda_{t+1}^{\text{join}}$ requires $O(n|\mathcal{I}|)$ time (the computations $\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\mu}$ and $\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}$ must be performed just once);
- Updating $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_{t+1})$ requires $O(|\mathcal{A}|)$ time.

In total, a single iteration requires $O(n|\mathcal{I}| + n|\mathcal{A}| + |\mathcal{A}|^2) = O(nd + |\mathcal{A}|^2)$ time. □

Regarding the number of iterations taken by the LARS algorithm, heuristically, it is on average $O(n)$ [RZ07]. This can be understood as follows. If $n > d$, it would take $O(d)$ steps to add all the variables. If $n < d$, then we only add at most n variables in the case of a unique solution, since $|\mathcal{A}| \leqslant \min\{n, d\}$. Dropping variables is rare as λ is successfully decreased, usually $O(1)$ times. In the worst case, though, the number of iterations can be at most $(3^d + 1)/2$ as previously mentioned.

4 Quantum algorithms

In this section, we introduce our quantum algorithms based on the LARS algorithm. In Section 4.1, we propose our simple quantum LARS algorithm that exactly computes the pathwise Lasso, while in Section 4.2 we improve upon this simple algorithm and propose the approximate quantum LARS algorithm. In Section 4.3, we dequantise the approximate LARS algorithm using sampling techniques. Finally, in Section 4.4, we analyse the algorithms' complexity for the case when $\mathbf{X}$ is a random matrix from the $\boldsymbol{\Sigma}$-Gaussian ensemble.

4.1 Simple quantum LARS algorithm

As a warm-up, we propose a simple quantum LARS algorithm for the Lasso path by quantising the search step of i_{t+1}^{join} and $\lambda_{t+1}^{\text{join}}$ using the quantum minimum-finding subroutine from Dürr and Høyer [DH96] (Fact 10). For this, we input the vectors $\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\mu}$ and $\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}$ into KP-tree data structures accessed by QRAMs. Moreover, as described in Section 2.3, we assume the matrix $\mathbf{X}$ is stored in a QROM. The final result is an improvement in the runtime over the usual LARS algorithm to $\widetilde{O}(n\sqrt{|\mathcal{I}|} + n|\mathcal{A}| + |\mathcal{A}|^2)$ per iteration.

Algorithm 2: Simple quantum LARS algorithm for the pathwise Lasso

Input: $T \in \mathbb{N}$, $\delta \in (0,1)$, $\mathbf{y} \in \mathbb{R}^n$, and $\mathbf{X} \in \mathbb{R}^{n\times d}$

Output: Exact regularisation path $\mathcal{P}$ with T kinks with probability $\geqslant 1-\delta$

Initialise $\mathcal{A} = \operatorname{argmax}_{j\in[d]} |\mathbf{X}_j^\top \mathbf{y}|$, $\mathcal{I} = [d]\backslash\mathcal{A}$, $\lambda_0 = \|\mathbf{X}^\top\mathbf{y}\|_\infty$, $\hat{\boldsymbol{\beta}}(\lambda_0) = \mathbf{0}$, $t = 0$

while $\mathcal{I} \neq \varnothing$, $t \leqslant T$ **do**

  $\boldsymbol{\eta}_{\mathcal{A}} \leftarrow \operatorname{sign}\left(\mathbf{X}_{\mathcal{A}}^\top(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t))\right)$

  $\boldsymbol{\mu} \leftarrow \mathbf{X}_{\mathcal{A}}^+\mathbf{y}$ // $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{A}|}$

  $\boldsymbol{\theta} \leftarrow (\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^+\boldsymbol{\eta}_{\mathcal{A}}$ // $\boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{A}|}$

  Compute $\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\mu}$ and $\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}$ classically and input them into QRAMs

  Define $\Lambda_i^{\text{join}} \triangleq \frac{\mathbf{X}_i^\top(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\boldsymbol{\mu})}{\pm 1 - \mathbf{X}_i^\top\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}}$ and $\Lambda_i^{\text{cross}} \triangleq \frac{\mu_i}{\theta_i} \cdot \mathbf{1}\left[\frac{\mu_i}{\theta_i} \leqslant \lambda_t\right]$

  $i_{t+1}^{\text{cross}} \leftarrow \operatorname{argmax}_{i\in\mathcal{A}}\{\Lambda_i^{\text{cross}}\}$ and $\lambda_{t+1}^{\text{cross}} \leftarrow \max_{i\in\mathcal{A}}\{\Lambda_i^{\text{cross}}\}$

  $i_{t+1}^{\text{join}} \leftarrow \operatorname{argmax}_{i\in\mathcal{I}}\{\Lambda_i^{\text{join}}\}$ and $\lambda_{t+1}^{\text{join}} \leftarrow \max_{i\in\mathcal{I}}\{\Lambda_i^{\text{join}}\}$ with failure probability $\frac{\delta}{T}$ (Fact 10)

  $\lambda_{t+1} \leftarrow \max\{\lambda_{t+1}^{\text{join}}, \lambda_{t+1}^{\text{cross}}\}$

  $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_{t+1}) \leftarrow \boldsymbol{\mu} - \lambda_{t+1}\boldsymbol{\theta}$

  if $\lambda_{t+1} = \lambda_{t+1}^{\text{cross}}$ **then**

    Move i_{t+1}^{cross} from $\mathcal{A}$ to $\mathcal{I}$

  else

    Move i_{t+1}^{join} from $\mathcal{I}$ to $\mathcal{A}$

  $t \leftarrow t+1$

return coefficients $[(\lambda_0, \hat{\boldsymbol{\beta}}(\lambda_0)), (\lambda_1, \hat{\boldsymbol{\beta}}(\lambda_1)), \dots]$

Theorem 22. *Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{y} \in \mathbb{R}^n$. Assume $\mathbf{X}$ is stored in a* QROM *and we have access to* QRAMs *of size $O(n)$. Let $\delta \in (0,1)$ and $T \in \mathbb{N}$. The simple quantum* LARS Algorithm 2 *outputs, with probability at least $1-\delta$, at most T kinks of the optimal regularisation path in time*

$$O\big(n\sqrt{|\mathcal{I}|}\log(T/\delta)\operatorname{poly}\log(nd) + n|\mathcal{A}| + |\mathcal{A}|^2\big)$$

per iteration, where $\mathcal{A}$ and $\mathcal{I}$ are the active and inactive sets of the corresponding iteration.

Proof. The time complexity of Algorithm 2 is basically the same as Algorithm 1, the only difference being that uploading $\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\mu}$ and $\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}$ into KP-trees takes $O(n\log n)$ time, and finding i_{t+1}^{join} and $\lambda_{t+1}^{\text{join}}$ now requires $O\big(n\sqrt{|\mathcal{I}|}\log(T/\delta)\operatorname{poly}\log(nd)\big)$ time, since that accessing $\mathbf{X}_i$ and computing $\Lambda_i^{\text{join}}(\mathcal{A}) = \frac{\mathbf{X}_i^\top(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\boldsymbol{\mu})}{\pm 1 - \mathbf{X}_i^\top\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}}$ requires $O(n\operatorname{poly}\log(nd))$ and $O(n)$ time, respectively. □

4.2 Approximate quantum LARS algorithm

We now improve our previous simple quantum LARS algorithm. The main idea is to *approximately* compute the joining times $\Lambda_i^{\text{join}}(\mathcal{A})$ (see Equation (11)) within the quantum minimum-finding algorithm instead of exactly computing them. This will lead to a quadratic improvement on the n dependence in the term $\widetilde{O}(n\sqrt{|\mathcal{I}|})$ to $\widetilde{O}(\sqrt{n|\mathcal{I}|})$. Such approximation, however, hinders our ability to exactly find the joining variables i_{t+1}^{join} and points $\lambda_{t+1}^{\text{join}}$. We can now only obtain a joining point $\widetilde{\lambda}_{t+1}^{\text{join}}$ which is ϵ-close to the true joining point $\lambda_{t+1}^{\text{join}}$. This imposes new complications on the correctness analysis of the LARS algorithm, since we can no longer guarantee that the KKT conditions are satisfied. In order to tackle this issue, we consider an approximate version of the KKT conditions. We note that a similar concept was already introduced in [MY12].

Definition 23. *Let $\epsilon \geqslant 0$. A vector $\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^d$ satisfies the $\mathsf{KKT}_\lambda(\epsilon)$ condition if and only if, $\forall j \in [d]$,*

$$\begin{aligned} \lambda(1-\epsilon) \leqslant \mathbf{X}_j^\top(\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}})\operatorname{sign}(\widetilde{\beta}_j) \leqslant \lambda \quad & \text{if } \widetilde{\beta}_j \neq 0, \\ |\mathbf{X}_j^\top(\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}})| \leqslant \lambda \quad & \text{if } \widetilde{\beta}_j = 0. \end{aligned}$$

The reason for introducing the above approximate version of the KKT conditions is that it leads to approximate Lasso solutions, as proven in the next lemma.

Lemma 24. *If a vector $\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^d$ satisfies the $\mathsf{KKT}_\lambda(\epsilon)$ condition for $\epsilon \geqslant 0$, then $\widetilde{\boldsymbol{\beta}}$ minimises the Lasso cost function $\mathcal{L}^{(\lambda)}(\boldsymbol{\beta})$ up to an error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$, i.e.,*

$$\frac{1}{2}\|\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}\|_2^2 + \lambda\|\widetilde{\boldsymbol{\beta}}\|_1 - \left(\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1\right) \leqslant \lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1.$$

Proof. The primal problem $\min_{\boldsymbol{\beta}\in\mathbb{R}^d, \mathbf{z}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{y} - \mathbf{z}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$ s.t. $\mathbf{z} = \mathbf{X}\boldsymbol{\beta}$, with dummy variable $\mathbf{z}$, leads to the dual problem

$$\max_{\boldsymbol{\kappa}\in\mathbb{R}^n} -\frac{1}{2}\boldsymbol{\kappa}^\top\boldsymbol{\kappa} - \boldsymbol{\kappa}^\top\mathbf{y} \quad \text{s.t. } \|\mathbf{X}^\top\boldsymbol{\kappa}\|_\infty \leqslant \lambda.$$

It is known that, given a pair of feasible primal and dual variables $(\widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\kappa}})$, the difference

$$\delta_\lambda(\widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\kappa}}) \triangleq \left(\frac{1}{2}\|\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}\|_2^2 + \lambda\|\widetilde{\boldsymbol{\beta}}\|_1\right) - \left(-\frac{1}{2}\widetilde{\boldsymbol{\kappa}}^\top\widetilde{\boldsymbol{\kappa}} - \widetilde{\boldsymbol{\kappa}}^\top\mathbf{y}\right)$$

is called a duality gap and provides an optimality guarantee (see, e.g., [BL06, Section 4.3]):

$$0 \leqslant \left(\frac{1}{2}\|\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}\|_2^2 + \lambda\|\widetilde{\boldsymbol{\beta}}\|_1\right) - \left(\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1\right) \leqslant \delta_\lambda(\widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\kappa}}).$$

The vector $\widetilde{\boldsymbol{\kappa}} = \mathbf{X}\widetilde{\boldsymbol{\beta}} - \mathbf{y}$ is feasible for the dual problem since $\widetilde{\boldsymbol{\beta}}$ satisfies the $\mathsf{KKT}_\lambda(\epsilon)$ condition ($\|\mathbf{X}^\top\widetilde{\boldsymbol{\kappa}}\|_\infty = \|\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}})\|_\infty \leqslant \lambda$). We can thus compute the duality gap,

$$\begin{aligned} \delta_\lambda(\widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\kappa}}) &= \left(\frac{1}{2}\|\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}\|_2^2 + \lambda\|\widetilde{\boldsymbol{\beta}}\|_1\right) - \left(-\frac{1}{2}\widetilde{\boldsymbol{\kappa}}^\top\widetilde{\boldsymbol{\kappa}} - \widetilde{\boldsymbol{\kappa}}^\top\mathbf{y}\right) \\ &= \frac{1}{2}\widetilde{\boldsymbol{\kappa}}^\top\widetilde{\boldsymbol{\kappa}} + \lambda\|\widetilde{\boldsymbol{\beta}}\|_1 + \frac{1}{2}\widetilde{\boldsymbol{\kappa}}^\top\widetilde{\boldsymbol{\kappa}} + \widetilde{\boldsymbol{\kappa}}^\top\mathbf{y} \\ &= \lambda\|\widetilde{\boldsymbol{\beta}}\|_1 + (\widetilde{\boldsymbol{\kappa}} + \mathbf{y})^\top\widetilde{\boldsymbol{\kappa}}, \end{aligned}$$

but

$$(\widetilde{\boldsymbol{\kappa}}+\mathbf{y})^\top\widetilde{\boldsymbol{\kappa}}=\widetilde{\boldsymbol{\beta}}^\top\mathbf{X}^\top(\mathbf{X}\widetilde{\boldsymbol{\beta}}-\mathbf{y})=\sum_{i=1}^{d}|\widetilde{\beta}_i|\mathbf{X}_i^\top(\mathbf{X}\widetilde{\boldsymbol{\beta}}-\mathbf{y})\operatorname{sign}(\widetilde{\beta}_i)\leqslant-\lambda(1-\epsilon)\|\widetilde{\boldsymbol{\beta}}\|_1,$$

using that $\widetilde{\boldsymbol{\beta}}$ satisfies the $\mathsf{KKT}_\lambda(\epsilon)$ condition. Therefore $\delta_\lambda(\widetilde{\boldsymbol{\beta}},\widetilde{\boldsymbol{\kappa}})\leqslant\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$. □

Remark 25. *Even though the error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$ might look unnatural, it can be relaxed to a relative error by observing that it implies*

$$\begin{aligned}\mathcal{L}^{(\lambda)}(\widetilde{\boldsymbol{\beta}})-\min_{\boldsymbol{\beta}\in\mathbb{R}^d}\mathcal{L}^{(\lambda)}(\boldsymbol{\beta})\leqslant\epsilon\lambda\|\widetilde{\boldsymbol{\beta}}\|_1\leqslant\epsilon\mathcal{L}^{(\lambda)}(\widetilde{\boldsymbol{\beta}})&\iff(1-\epsilon)\big(\mathcal{L}^{(\lambda)}(\widetilde{\boldsymbol{\beta}})-\min_{\boldsymbol{\beta}\in\mathbb{R}^d}\mathcal{L}^{(\lambda)}(\boldsymbol{\beta})\big)\leqslant\epsilon\min_{\boldsymbol{\beta}\in\mathbb{R}^d}\mathcal{L}^{(\lambda)}(\boldsymbol{\beta})\\&\iff\mathcal{L}^{(\lambda)}(\widetilde{\boldsymbol{\beta}})-\min_{\boldsymbol{\beta}\in\mathbb{R}^d}\mathcal{L}^{(\lambda)}(\boldsymbol{\beta})\leqslant\frac{\epsilon}{1-\epsilon}\min_{\boldsymbol{\beta}\in\mathbb{R}^d}\mathcal{L}^{(\lambda)}(\boldsymbol{\beta}).\end{aligned}$$

We now present our approximate quantum LARS Algorithm 3 for the pathwise Lasso. Its main idea is quite simple: we improve the search of the joining variable i_{t+1}^{join} and time $\lambda_{t+1}^{\text{join}}$ over $i\in\mathcal{I}$ by approximately computing the joining times $\Lambda_i^{\text{join}}(\mathcal{A})=\frac{\mathbf{X}_i^\top(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\boldsymbol{\mu})}{\pm1-\mathbf{X}_i^\top\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}}$ and using the approximate quantum minimum-finding subroutine from Chen and de Wolf [CdW23] (Fact 11). More specifically, the joining variable $\widetilde{i}_{t+1}^{\text{join}}$ is obtained to be any variable $j\in\mathcal{I}$ that maximises $\Lambda_j^{\text{join}}(\mathcal{A})$ up to some small relative error ϵ, i.e., at any iteration, Algorithm 3 randomly samples from the set

$$\left\{j\in\mathcal{I}:\Lambda_j^{\text{join}}(\mathcal{A})\geqslant\left(1-\frac{\epsilon}{1+\alpha_{\mathcal{A}}}\right)\max_{i\in\mathcal{I}}\{\Lambda_i^{\text{join}}(\mathcal{A})\}\right\},\tag{12}$$

where $\alpha_{\mathcal{A}}>0$ is such that $\|\mathbf{X}_{\mathcal{A}}^+\mathbf{X}_{\mathcal{A}^c}\|_1\leqslant\alpha_{\mathcal{A}}$. Since the corresponding joining time $\Lambda_{\widetilde{i}_{t+1}^{\text{join}}}^{\text{join}}(\mathcal{A})$ can be smaller than the true joining time $\lambda_{t+1}^{\text{join}}=\max_{i\in\mathcal{I}}\{\Lambda_i^{\text{join}}(\mathcal{A})\}$, we rescale it by a factor $\left(1-\frac{\epsilon}{1+\alpha_{\mathcal{A}}}\right)^{-1}$ and set $\widetilde{\lambda}_{t+1}^{\text{join}}=\left(1-\frac{\epsilon}{1+\alpha_{\mathcal{A}}}\right)^{-1}\Lambda_{\widetilde{i}_{t+1}^{\text{join}}}^{\text{join}}(\mathcal{A})$. This ensures that $\widetilde{\lambda}_{t+1}^{\text{join}}\geqslant\lambda_{t+1}^{\text{join}}$ and consequently that the current iteration stops before $\lambda_{t+1}^{\text{join}}$, which guarantees that the approximate KKT conditions are satisfied as shown in Theorem 26 below.

We notice that, since the joining variable is randomly sampled from the set in Equation (12), it might be possible that $\widetilde{\lambda}_{t+2}^{\text{join}}>\widetilde{\lambda}_{t+1}^{\text{join}}$, i.e., the joining time at some later iteration is greater than the joining time at some earlier iteration. While this is counter to the original LARS algorithm, where the regularisation parameter λ always decreases, there is in principle no problem in allowing λ to increase within a small interval as long as the approximate KKT conditions are satisfied. Nonetheless, in order to avoid redundant segments in the Lasso path, we set the next iteration's regularisation parameter λ_{t+1} as the minimum between the current λ_t and $\max\{\lambda_{t+1}^{\text{cross}},\widetilde{\lambda}_{t+1}^{\text{join}}\}$. The solution path can thus stay "stationary" for a few iterations. Ideally, one could just move all the variables from the set in Equation (12) into $\mathcal{A}$ at once and the result would be the same. This is reminiscent to the approximate homotopy algorithm of Mairal and Yu [MY12].

We show next that the imprecision in finding $\lambda_{t+1}^{\text{join}}$ leads to a path that satisfies the approximate KKT conditions from Definition 23 and thus approximates the Lasso function up to a small error.

Theorem 26. *Let $T\in\mathbb{N}$, $\epsilon\geqslant0$, $\mathbf{X}\in\mathbb{R}^{n\times d}$, and $\mathbf{y}\in\mathbb{R}^n$. For $\mathcal{A}\subseteq[d]$ of size $|\mathcal{A}|\leqslant T$, let $\alpha_{\mathcal{A}}>0$ be such that $\|\mathbf{X}_{\mathcal{A}}^+\mathbf{X}_{\mathcal{A}^c}\|_1\leqslant\alpha_{\mathcal{A}}$. Consider an approximate* LARS *algorithm (e.g., Algorithm 3) that returns a solution path $\widetilde{\mathcal{P}}=\{\widetilde{\boldsymbol{\beta}}(\lambda)\in\mathbb{R}^d:\lambda>0\}$ with at most T kinks and wherein, at each iteration t, the joining variable $\widetilde{i}_{t+1}^{\text{join}}$ is taken from the set*

$$\left\{j\in\mathcal{I}:\Lambda_j^{\text{join}}(\mathcal{A})\geqslant\left(1-\frac{\epsilon}{1+\alpha_{\mathcal{A}}}\right)\max_{i\in\mathcal{I}}\{\Lambda_i^{\text{join}}(\mathcal{A})\}\right\}$$

Algorithm 3: Approximate quantum LARS algorithm for the pathwise Lasso

Input: $T \in \mathbb{N}$, $\delta \in (0,1)$, $\epsilon > 0$, $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n\times d}$, and $\{\alpha_{\mathcal{A}} \geqslant \|\mathbf{X}_{\mathcal{A}}^{+}\mathbf{X}_{\mathcal{A}^c}\|_1\}_{\mathcal{A}\subseteq[d]:|\mathcal{A}|\leqslant T}$

Output: Regularisation path $\widetilde{\mathcal{P}}$ with T kinks and error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}(\lambda)\|_1$ with probability $\geqslant 1-\delta$

Initialise $\mathcal{A} = \operatorname{argmax}_{j\in[d]} |\mathbf{X}_j^\top \mathbf{y}|$, $\mathcal{I} = [d]\backslash\mathcal{A}$, $\lambda_0 = \|\mathbf{X}^\top\mathbf{y}\|_\infty$, $\widetilde{\boldsymbol{\beta}}(\lambda_0) = 0$, $t = 0$

while $\mathcal{I} \neq \varnothing$, $t \leqslant T$ **do**

  $\boldsymbol{\eta}_{\mathcal{A}} \leftarrow \frac{1}{\lambda_t}\mathbf{X}_{\mathcal{A}}^\top(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t))$

  $\boldsymbol{\mu} \leftarrow \mathbf{X}_{\mathcal{A}}^{+}\mathbf{y}$ // $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{A}|}$

  $\boldsymbol{\theta} \leftarrow (\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{+}\boldsymbol{\eta}_{\mathcal{A}}$ // $\boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{A}|}$

  Classically compute $\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\mu}$ and $\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}$ and input them into QRAMs and KP-trees

  Define $\Lambda_i^{\text{join}} \triangleq \frac{\mathbf{X}_i^\top(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\boldsymbol{\mu})}{\pm 1 - \mathbf{X}_i^\top\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}}$ and $\Lambda_i^{\text{cross}} \triangleq \frac{\mu_i}{\theta_i}\cdot\mathbf{1}\left[\frac{\mu_i}{\theta_i} \leqslant \lambda_t\right]$

  $i_{t+1}^{\text{cross}} \leftarrow \operatorname{argmax}_{i\in\mathcal{A}}\{\Lambda_i^{\text{cross}}\}$ and $\lambda_{t+1}^{\text{cross}} \leftarrow \max_{i\in\mathcal{A}}\{\Lambda_i^{\text{cross}}\}$

  Obtain $\widetilde{i}_{t+1}^{\text{join}} \in \left\{j \in \mathcal{I} : \Lambda_j^{\text{join}} \geqslant \left(1 - \frac{\epsilon}{1+\alpha_{\mathcal{A}}}\right)\max_{i\in\mathcal{I}}\{\Lambda_i^{\text{join}}\}\right\}$ with failure probability $\frac{\delta}{T}$ (Fact 11 and Lemma 13)

  $\widetilde{\lambda}_{t+1}^{\text{join}} \leftarrow \left(1 - \frac{\epsilon}{1+\alpha_{\mathcal{A}}}\right)^{-1}\Lambda_{\widetilde{i}_{t+1}^{\text{join}}}^{\text{join}}$

  $\lambda_{t+1} \leftarrow \min\{\lambda_t, \max\{\lambda_{t+1}^{\text{cross}}, \widetilde{\lambda}_{t+1}^{\text{join}}\}\}$

  $\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_{t+1}) \leftarrow \boldsymbol{\mu} - \lambda_{t+1}\boldsymbol{\theta}$

  if $\lambda_{t+1} = \lambda_{t+1}^{\text{cross}}$ **then**

    Move i_{t+1}^{cross} from $\mathcal{A}$ to $\mathcal{I}$

  else

    Move $\widetilde{i}_{t+1}^{\text{join}}$ from $\mathcal{I}$ to $\mathcal{A}$

  $t \leftarrow t+1$

return coefficients $[(\lambda_0, \widetilde{\boldsymbol{\beta}}(\lambda_0)), (\lambda_1, \widetilde{\boldsymbol{\beta}}(\lambda_1)), \dots]$

and the corresponding joining time $\widetilde{\lambda}_{t+1}^{\text{join}}$ *is*

$$\widetilde{\lambda}_{t+1}^{\text{join}} = \left(1 - \frac{\epsilon}{1+\alpha_{\mathcal{A}}}\right)^{-1}\Lambda_{\widetilde{i}_{t+1}^{\text{join}}}^{\text{join}}(\mathcal{A}),$$

where $\mathcal{A}$ *and* $\mathcal{I}$ *are the active and inactive sets at the corresponding iteration* t *and*

$$\Lambda_j^{\text{join}}(\mathcal{A}) = \frac{\mathbf{X}_j^\top(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\boldsymbol{\mu})}{\pm 1 - \mathbf{X}_j^\top\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}}, \quad \boldsymbol{\mu} = \mathbf{X}_{\mathcal{A}}^{+}\mathbf{y}, \quad \boldsymbol{\theta} = (\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{+}\boldsymbol{\eta}_{\mathcal{A}}, \quad \boldsymbol{\eta}_{\mathcal{A}} = \frac{1}{\lambda_t}\mathbf{X}_{\mathcal{A}}^\top(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t)).$$

Then $\widetilde{\mathcal{P}}$ *is a continuous and piecewise linear approximate regularisation path with error* $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}(\lambda)\|_1$.

Proof. We shall prove that the Lasso solution $\widetilde{\boldsymbol{\beta}}(\lambda)$ satisfies the $\mathsf{KKT}_\lambda(\epsilon)$ condition for all $\lambda > 0$. According to Lemma 24, $\widetilde{\mathcal{P}}$ is then an approximate regularisation path with error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}(\lambda)\|_1$.

The proof is by induction on the iteration loop t. The case $t = 0$ is trivial as $\hat{\boldsymbol{\beta}}(\lambda) = \mathbf{0}$ for $\lambda \geqslant \lambda_0$. Assume then that the computed path through iteration $t-1$ satisfies $\mathsf{KKT}_\lambda(\epsilon)$ for all $\lambda \geqslant \lambda_t$. Consider the t-th iteration with active set $\mathcal{A} = \{i \in [d] : \widetilde{\beta}_i \neq 0\}$ and equicorrelation signs $\boldsymbol{\eta}_{\mathcal{A}} = \frac{1}{\lambda_t}\mathbf{X}_{\mathcal{A}}^\top(\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}(\lambda_t))$. The induction hypothesis implies that the current Lasso solution $\widetilde{\boldsymbol{\beta}}(\lambda_t)$ satisfies $\mathsf{KKT}_{\lambda_t}(\epsilon)$ at λ_t:

$$\begin{aligned} \lambda_t(1-\epsilon) \leqslant \mathbf{X}_j^\top(\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}(\lambda_t))\operatorname{sign}(\widetilde{\beta}_j(\lambda_t)) \leqslant \lambda_t \quad &\text{if } \widetilde{\beta}_j \neq 0, \\ |\mathbf{X}_j^\top(\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}(\lambda_t))| \leqslant \lambda_t \quad &\text{if } \widetilde{\beta}_j = 0. \end{aligned}$$

Recall that for $\lambda \leqslant \lambda_t$ the Lasso solution is

$$\widetilde{\boldsymbol{\beta}}_{\mathcal{I}}(\lambda) = \mathbf{0} \quad \text{and} \quad \widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda) = \mathbf{X}_{\mathcal{A}}^{+}(\mathbf{y} - \lambda(\mathbf{X}_{\mathcal{A}}^{+})^{\top}\boldsymbol{\eta}_{\mathcal{A}}).$$

Thus

$$\begin{aligned}
\mathbf{X}_{\mathcal{A}}^{\top}(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda)) &= \mathbf{X}_{\mathcal{A}}^{\top}\mathbf{y} - \mathbf{X}_{\mathcal{A}}^{\top}\mathbf{X}_{\mathcal{A}}\mathbf{X}_{\mathcal{A}}^{+}\mathbf{y} + \lambda\mathbf{X}_{\mathcal{A}}^{\top}\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^{\top}\mathbf{X}_{\mathcal{A}})^{+}\boldsymbol{\eta}_{\mathcal{A}} \\
&= \lambda\mathbf{X}_{\mathcal{A}}^{\top}(\mathbf{X}_{\mathcal{A}}^{\top})^{+}\boldsymbol{\eta}_{\mathcal{A}} && (\text{by } \mathbf{X}_{\mathcal{A}}^{\top}\mathbf{X}_{\mathcal{A}}\mathbf{X}_{\mathcal{A}}^{+} = \mathbf{X}_{\mathcal{A}}^{\top}) \\
&= \lambda\boldsymbol{\eta}_{\mathcal{A}}, && (\text{by } \boldsymbol{\eta}_{\mathcal{A}} \in \operatorname{row}(\mathbf{X}_{\mathcal{A}}))
\end{aligned}$$

where we used the identity $\mathbf{A}^{\top}\mathbf{A}\mathbf{A}^{+} = \mathbf{A}^{\top}$. Since

$$\eta_j \operatorname{sign}(\widetilde{\beta}_j(\lambda)) = \eta_j \operatorname{sign}(\widetilde{\beta}_j(\lambda_t)) = \frac{1}{\lambda_t}\mathbf{X}_j^{\top}(\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}(\lambda_t)) \operatorname{sign}(\widetilde{\beta}_j(\lambda_t))$$

for λ sufficiently close to λ_t and since $\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t)$ satisfies $\mathsf{KKT}_{\lambda_t}(\epsilon)$, we conclude that

$$\lambda(1-\epsilon) \leqslant \mathbf{X}_j^{\top}(\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}(\lambda)) \operatorname{sign}(\widetilde{\beta}_j(\lambda)) \leqslant \lambda \quad \text{for all } j \in \mathcal{A}$$

as required. As λ decreases, one of the following conditions must break: either $\eta_j \neq 0$ for all $j \in \mathcal{A}$ or $\|\mathbf{X}_{\mathcal{I}}^{\top}(\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}(\lambda))\|_{\infty} \leqslant \lambda$. The first breaks at the next crossing time $\lambda_{t+1}^{\text{cross}} = \max_{i\in\mathcal{A}}\{\Lambda_i^{\text{cross}}(\mathcal{A})\}$, and the second breaks at the next joining time $\lambda_{t+1}^{\text{join}} = \max_{i\in\mathcal{I}}\{\Lambda_i^{\text{join}}(\mathcal{A})\}$. Since we only decrease λ to $\lambda_{t+1} = \min\{\lambda_t, \max\{\lambda_{t+1}^{\text{cross}}, \widetilde{\lambda}_{t+1}^{\text{join}}\}\}$, we have verified that $\widetilde{\boldsymbol{\beta}}(\lambda)$ satisfies $\mathsf{KKT}_{\lambda}(\epsilon)$ for $\lambda \geqslant \lambda_{t+1}$.

We now prove that adding or deleting variables from the active set preserves the condition $\mathsf{KKT}_{\lambda_{t+1}}(\epsilon)$ at λ_{t+1}. More specifically, let $\mathcal{A}^*$ and $\boldsymbol{\eta}_{\mathcal{A}^*}^*$ denote the active set and equicorrelation signs at the beginning of the $(t+1)$-th iteration.

Case 1 (Deletion): Let us start with the case when a variable leaves the active set at $\lambda_{t+1} = \lambda_{t+1}^{\text{cross}}$, i.e., $\mathcal{A}^* = \mathcal{A}\backslash\{i_{t+1}^{\text{cross}}\}$ is formed by removing an element from $\mathcal{A}$. The Lasso solution before deletion with equicorrelation signs $\boldsymbol{\eta}_{\mathcal{A}}$ is

$$\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_{t+1}) = \begin{bmatrix} \widetilde{\boldsymbol{\beta}}_{\mathcal{A}^*}(\lambda_{t+1}) \\ \widetilde{\beta}_{i_{t+1}^{\text{cross}}}(\lambda_{t+1}) \end{bmatrix} = \begin{bmatrix} [\mathbf{X}_{\mathcal{A}}^{+}(\mathbf{y} - \lambda_{t+1}(\mathbf{X}_{\mathcal{A}}^{+})^{\top}\boldsymbol{\eta}_{\mathcal{A}})]_{\mathcal{A}^*} \\ 0 \end{bmatrix}$$

since the variable i_{t+1}^{cross} crosses through zero at $\lambda_{t+1} = \lambda_{t+1}^{\text{cross}}$. On the other hand, the Lasso solution after deletion is

$$\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}^*(\lambda_{t+1}) = \begin{bmatrix} \widetilde{\boldsymbol{\beta}}_{\mathcal{A}^*}^*(\lambda_{t+1}) \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{\mathcal{A}^*}^{+}(\mathbf{y} - \lambda_{t+1}(\mathbf{X}_{\mathcal{A}^*}^{+})^{\top}\boldsymbol{\eta}_{\mathcal{A}^*}^*) \\ 0 \end{bmatrix},$$

where $\boldsymbol{\eta}_{\mathcal{A}^*}^* = \frac{1}{\lambda_{t+1}}\mathbf{X}_{\mathcal{A}^*}^{\top}(\mathbf{y} - \mathbf{X}_{\mathcal{A}^*}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}^*}(\lambda_{t+1}))$. Therefore

$$\begin{aligned}
\mathbf{X}_{\mathcal{A}^*}^{+}(\mathbf{y} - \lambda_{t+1}(\mathbf{X}_{\mathcal{A}^*}^{+})^{\top}\boldsymbol{\eta}_{\mathcal{A}^*}^*) &= \mathbf{X}_{\mathcal{A}^*}^{+}\mathbf{y} - \mathbf{X}_{\mathcal{A}^*}^{+}(\mathbf{X}_{\mathcal{A}^*}^{+})^{\top}\mathbf{X}_{\mathcal{A}^*}^{\top}\mathbf{y} + \mathbf{X}_{\mathcal{A}^*}^{+}(\mathbf{X}_{\mathcal{A}^*}^{+})^{\top}\mathbf{X}_{\mathcal{A}^*}^{\top}\mathbf{X}_{\mathcal{A}^*}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}^*}(\lambda_{t+1}) \\
&= \mathbf{X}_{\mathcal{A}^*}^{+}\mathbf{X}_{\mathcal{A}^*}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}^*}(\lambda_{t+1}), && (\text{by } \mathbf{X}_{\mathcal{A}^*}^{+}(\mathbf{X}_{\mathcal{A}^*}^{+})^{\top}\mathbf{X}_{\mathcal{A}^*}^{\top} = \mathbf{X}_{\mathcal{A}^*}^{+})
\end{aligned}$$

again using that $\mathbf{A}^{+}(\mathbf{A}^{+})^{\top}\mathbf{A}^{\top} = \mathbf{A}^{+}$. The solution $\widetilde{\boldsymbol{\beta}}_{\mathcal{A}^*}(\lambda_{t+1})$ to the above equation must be

$$\widetilde{\boldsymbol{\beta}}_{\mathcal{A}^*}(\lambda_{t+1}) = \mathbf{X}_{\mathcal{A}^*}^{+}(\mathbf{y} - \lambda_{t+1}(\mathbf{X}_{\mathcal{A}^*}^{+})^{\top}\boldsymbol{\eta}_{\mathcal{A}^*}^*) + \mathbf{b} = \widetilde{\boldsymbol{\beta}}_{\mathcal{A}^*}^*(\lambda_{t+1}) + \mathbf{b},$$

where $\mathbf{b} \in \operatorname{null}(\mathbf{X}_{\mathcal{A}^*})$. Since $\widetilde{\boldsymbol{\beta}}_{\mathcal{A}^*}(\lambda_{t+1})$ must have minimum ℓ_2-norm, $\mathbf{b} = \mathbf{0}$ and so

$$\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}^*(\lambda_{t+1}) = \begin{bmatrix} \widetilde{\boldsymbol{\beta}}_{\mathcal{A}^*}^*(\lambda_{t+1}) \\ 0 \end{bmatrix} = \begin{bmatrix} \widetilde{\boldsymbol{\beta}}_{\mathcal{A}^*}(\lambda_{t+1}) \\ 0 \end{bmatrix},$$

which implies that $\widetilde{\mathcal{P}}$ is continuous at λ_{t+1}. Finally,

$$\mathbf{X}_j^\top(\mathbf{y} - \mathbf{X}_{\mathcal{A}*}\widetilde{\boldsymbol{\beta}}^*_{\mathcal{A}*}(\lambda_{t+1})) = \mathbf{X}_j^\top(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_{t+1})) \quad \text{for all } j \in [d]$$

and so $\widetilde{\boldsymbol{\beta}}^*(\lambda_{t+1})$ satisfies the $\mathsf{KKT}_{\lambda_{t+1}}(\epsilon)$ condition.

Case 2 (Insertion): Now we look at the case when a variable joins the active set at $\lambda_{t+1} = \min\{\lambda_t, \widetilde{\lambda}^{\text{join}}_{t+1}\}$, i.e., $\mathcal{A}^* = \mathcal{A} \cup \{\widetilde{i}^{\text{join}}_{t+1}\}$ is formed by adding an element to $\mathcal{A}$. The Lasso solution before insertion is

$$\widetilde{\boldsymbol{\beta}}_{\mathcal{A}*}(\lambda_{t+1}) = \begin{bmatrix} \widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_{t+1}) \\ 0 \end{bmatrix},$$

while the Lasso solution after insertion with equicorrelation signs

$$\boldsymbol{\eta}^*_{\mathcal{A}*} = \frac{1}{\lambda_{t+1}}\mathbf{X}_{\mathcal{A}*}^\top(\mathbf{y} - \mathbf{X}_{\mathcal{A}*}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}*}(\lambda_{t+1})) = \frac{1}{\lambda_{t+1}}\mathbf{X}_{\mathcal{A}*}^\top(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_{t+1}))$$

is

$$\begin{aligned}
\widetilde{\boldsymbol{\beta}}^*_{\mathcal{A}*}(\lambda_{t+1}) &= \mathbf{X}^+_{\mathcal{A}*}(\mathbf{y} - \lambda_{t+1}(\mathbf{X}^+_{\mathcal{A}*})^\top\boldsymbol{\eta}^*_{\mathcal{A}*}) \\
&= \mathbf{X}^+_{\mathcal{A}*}\mathbf{y} - \mathbf{X}^+_{\mathcal{A}*}(\mathbf{X}^+_{\mathcal{A}*})^\top\mathbf{X}_{\mathcal{A}*}^\top\mathbf{y} + \mathbf{X}^+_{\mathcal{A}*}(\mathbf{X}^+_{\mathcal{A}*})^\top\mathbf{X}_{\mathcal{A}*}^\top\mathbf{X}_{\mathcal{A}*}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}*}(\lambda_{t+1}) \\
&= \mathbf{X}^+_{\mathcal{A}*}\mathbf{X}_{\mathcal{A}*}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}*}(\lambda_{t+1}), \qquad (\text{by } \mathbf{X}^+_{\mathcal{A}*}(\mathbf{X}^+_{\mathcal{A}*})^\top\mathbf{X}_{\mathcal{A}*}^\top = \mathbf{X}^+_{\mathcal{A}*})
\end{aligned}$$

again using that $\mathbf{A}^+(\mathbf{A}^+)^\top\mathbf{A}^\top = \mathbf{A}^+$. The solution $\widetilde{\boldsymbol{\beta}}_{\mathcal{A}*}(\lambda_{t+1})$ to the above equation must be

$$\widetilde{\boldsymbol{\beta}}_{\mathcal{A}*}(\lambda_{t+1}) = \mathbf{X}^+_{\mathcal{A}*}(\mathbf{y} - \lambda_{t+1}(\mathbf{X}^+_{\mathcal{A}*})^\top\boldsymbol{\eta}^*_{\mathcal{A}*}) + \mathbf{b} = \widetilde{\boldsymbol{\beta}}^*_{\mathcal{A}*}(\lambda_{t+1}) + \mathbf{b},$$

where $\mathbf{b} \in \text{null}(\mathbf{X}_{\mathcal{A}*})$. Since $\widetilde{\boldsymbol{\beta}}_{\mathcal{A}*}(\lambda_{t+1})$ must have minimum ℓ_2-norm, $\mathbf{b} = \mathbf{0}$ and so

$$\widetilde{\boldsymbol{\beta}}^*_{\mathcal{A}*}(\lambda_{t+1}) = \widetilde{\boldsymbol{\beta}}_{\mathcal{A}*}(\lambda_{t+1}) = \begin{bmatrix} \widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_{t+1}) \\ 0 \end{bmatrix},$$

which implies that $\widetilde{\mathcal{P}}$ is continuous at λ_{t+1}. Also,

$$\mathbf{X}_j^\top(\mathbf{y} - \mathbf{X}_{\mathcal{A}*}\widetilde{\boldsymbol{\beta}}^*_{\mathcal{A}*}(\lambda_{t+1})) = \mathbf{X}_j^\top(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_{t+1})) \quad \text{for all } j \in [d].$$

It only remains to prove that the variable $\widetilde{i}^{\text{join}}_{t+1}$ satisfies the $\mathsf{KKT}_{\lambda_{t+1}}(\epsilon)$ condition once it is added to $\mathcal{A}$. Indeed, first notice that the variable $\widetilde{i}^{\text{join}}_{t+1}$ achieves correlation

$$\left|\mathbf{X}^\top_{\widetilde{i}^{\text{join}}_{t+1}}(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}}))\right| = \Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}}$$

at the point $\Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}} = \Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}}(\mathcal{A})$. Define $\Delta^{\text{join}}_{t+1} \triangleq \min\{\lambda_t, \widetilde{\lambda}^{\text{join}}_{t+1}\} - \Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}}$ and note that $\Delta^{\text{join}}_{t+1} \leqslant \frac{\epsilon}{1+\alpha_{\mathcal{A}}}\widetilde{\lambda}^{\text{join}}_{t+1}$

since $\lambda_t \geqslant \Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}}$ and $\big(1-\frac{\epsilon}{1+\alpha_{\mathcal{A}}}\big)\widetilde{\lambda}^{\text{join}}_{t+1} \leqslant \Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}}$. Then, at the point $\lambda_{t+1} = \min\{\lambda_t, \widetilde{\lambda}^{\text{join}}_{t+1}\}$,

$$
\begin{aligned}
\big|\mathbf{X}^\top_{\widetilde{i}^{\text{join}}_{t+1}}(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\widetilde{\lambda}^{\text{join}}_{t+1}))\big| &= \big|\mathbf{X}^\top_{\widetilde{i}^{\text{join}}_{t+1}}(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}})) + \Delta^{\text{join}}_{t+1}\mathbf{X}^\top_{\widetilde{i}^{\text{join}}_{t+1}}(\mathbf{X}^+_{\mathcal{A}})^\top\boldsymbol{\eta}_{\mathcal{A}}\big| \\
&\geqslant \big|\mathbf{X}^\top_{\widetilde{i}^{\text{join}}_{t+1}}(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}}))\big| - \Delta^{\text{join}}_{t+1}\big|\mathbf{X}^\top_{\widetilde{i}^{\text{join}}_{t+1}}(\mathbf{X}^+_{\mathcal{A}})^\top\boldsymbol{\eta}_{\mathcal{A}}\big| \\
&\hspace{18em}\text{(Triangle inequality)} \\
&= \Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}} - \Delta^{\text{join}}_{t+1}\big|\mathbf{X}^\top_{\widetilde{i}^{\text{join}}_{t+1}}(\mathbf{X}^+_{\mathcal{A}})^\top\boldsymbol{\eta}_{\mathcal{A}}\big| \qquad \big(\big|\mathbf{X}^\top_{\widetilde{i}^{\text{join}}_{t+1}}(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}}))\big| = \Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}}\big) \\
&= \Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}} - \frac{\Delta^{\text{join}}_{t+1}}{\lambda_t}\big|\mathbf{X}^\top_{\widetilde{i}^{\text{join}}_{t+1}}(\mathbf{X}^+_{\mathcal{A}})^\top\mathbf{X}^\top_{\mathcal{A}}(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t))\big| \\
&\geqslant \Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}} - \frac{\Delta^{\text{join}}_{t+1}}{\lambda_t}\|\mathbf{X}^+_{\mathcal{A}}\mathbf{X}_{\widetilde{i}^{\text{join}}_{t+1}}\|_1\|\mathbf{X}^\top_{\mathcal{A}}(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t))\|_\infty \\
&\hspace{18em}\text{(Hölder's inequality)} \\
&\geqslant \Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}} - \Delta^{\text{join}}_{t+1}\|\mathbf{X}^+_{\mathcal{A}}\mathbf{X}_{\widetilde{i}^{\text{join}}_{t+1}}\|_1 \qquad (\|\mathbf{X}^\top_{\mathcal{A}}(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t))\|_\infty \leqslant \lambda_t) \\
&\geqslant \Big(1-\frac{\epsilon}{1+\alpha_{\mathcal{A}}}\Big)\widetilde{\lambda}^{\text{join}}_{t+1} - \frac{\epsilon\widetilde{\lambda}^{\text{join}}_{t+1}}{1+\alpha_{\mathcal{A}}}\|\mathbf{X}^+_{\mathcal{A}}\mathbf{X}_{\widetilde{i}^{\text{join}}_{t+1}}\|_1 \quad \big(\Lambda^{\text{join}}_{\widetilde{i}^{\text{join}}_{t+1}} \geqslant \big(1-\tfrac{\epsilon}{1+\alpha_{\mathcal{A}}}\big)\widetilde{\lambda}^{\text{join}}_{t+1}\big) \\
&\geqslant (1-\epsilon)\widetilde{\lambda}^{\text{join}}_{t+1}, \qquad (\textstyle\max_{j\in\mathcal{A}^c}\|\mathbf{X}^+_{\mathcal{A}}\mathbf{X}_j\|_1 \leqslant \alpha_{\mathcal{A}})
\end{aligned}
$$

where we used that $\widetilde{\boldsymbol{\beta}}(\lambda_t)$ satisfies the $\mathsf{KKT}_{\lambda_t}(\epsilon)$ condition at λ_t, $\|\mathbf{X}^\top_{\mathcal{A}}(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t))\|_\infty \leqslant \lambda_t$. ☐

After asserting the correctness of Algorithm 3, we now analyse its complexity. Specifically, we show how to sample the joining variable $\widetilde{i}^{\text{join}}_{t+1}$ from the set in Equation (12) by combining the approximate quantum minimum-finding subroutine from Chen and de Wolf [CdW23] (Fact 11) and the unitary map that approximates (parts of) the joining times $\Lambda^{\text{join}}_i(\mathcal{A}) = \frac{\mathbf{X}^\top_i(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\boldsymbol{\mu})}{\pm 1-\mathbf{X}^\top_i\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}}$ (Lemma 13). Our final complexity depends on the mutual incoherence between different column subspaces and the overlap between $\mathbf{y}$ and different column subspaces.

Theorem 27. *Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{y} \in \mathbb{R}^n$. Assume $\mathbf{X}$ is stored in a* QROM *and we have access to* QRAMs *of size $O(n)$. Let $\delta \in (0,1)$, $\epsilon > 0$, and $T \in \mathbb{N}$. For $\mathcal{A} \subseteq [d]$ of size $|\mathcal{A}| \leqslant T$, assume there are $\alpha_{\mathcal{A}}, \gamma_{\mathcal{A}} \in (0,1)$ such that*

$$
\|\mathbf{X}^+_{\mathcal{A}}\mathbf{X}_{\mathcal{A}^c}\|_1 \leqslant \alpha_{\mathcal{A}} \quad \textit{and} \quad \frac{\|\mathbf{X}^\top(\mathbf{I}-\mathbf{X}_{\mathcal{A}}\mathbf{X}^+_{\mathcal{A}})\mathbf{y}\|_\infty}{\|\mathbf{X}\|_{\max}\|(\mathbf{I}-\mathbf{X}_{\mathcal{A}}\mathbf{X}^+_{\mathcal{A}})\mathbf{y}\|_1} \geqslant \gamma_{\mathcal{A}}.
$$

The approximate quantum LARS *Algorithm 3 outputs, with probability at least $1-\delta$, a continuous piecewise linear approximate regularisation path $\widetilde{\mathcal{P}} = \{\widetilde{\boldsymbol{\beta}}(\lambda) \in \mathbb{R}^d : \lambda > 0\}$ with error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}(\lambda)\|_1$ and at most T kinks in time*

$$
O\left(\frac{\gamma^{-1}_{\mathcal{A}} + \sqrt{n}\|\mathbf{X}\|_{\max}\|\mathbf{X}^+_{\mathcal{A}}\|_2}{(1-\alpha_{\mathcal{A}})\epsilon}\sqrt{|\mathcal{I}|}\log^2(T/\delta)\operatorname{poly}\log(nd) + n|\mathcal{A}| + |\mathcal{A}|^2\right)
$$

per iteration, where $\mathcal{A}$ and $\mathcal{I}$ are the active and inactive sets of the corresponding iteration.

Proof. The correctness of the approximate quantum LARS Algorithm 3 follows from Theorem 26, since it is a LARS algorithm wherein the joining variable $\widetilde{i}^{\text{join}}_{t+1}$ is taken from the set

$$
\left\{j \in \mathcal{I} : \Lambda^{\text{join}}_j(\mathcal{A}) \geqslant \left(1-\frac{\epsilon}{1+\alpha_{\mathcal{A}}}\right)\max_{i\in\mathcal{I}}\{\Lambda^{\text{join}}_i(\mathcal{A})\}\right\} \tag{13}
$$

and the corresponding joining point $\widetilde{\lambda}_{t+1}^{\text{join}}$ is

$$\widetilde{\lambda}_{t+1}^{\text{join}} = \left(1 - \frac{\epsilon}{1+\alpha_{\mathcal{A}}}\right)^{-1} \Lambda^{\text{join}}_{\widetilde{i}_{t+1}^{\text{join}}}(\mathcal{A}).$$

Regarding the time complexity of the algorithm, the most expensive steps are computing $\mathbf{X}_{\mathcal{A}}^{+}$ in time $O(n|\mathcal{A}| + |\mathcal{A}|^2)$, inputting $\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\mu}$ and $\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}$ into KP-trees in time $O(n \log n)$, and sampling $\widetilde{i}_{t+1}^{\text{join}}$. We now show how to obtain the joining variable $\widetilde{i}_{t+1}^{\text{join}}$, i.e., how to sample it from the set in Equation (13). In the following, fix $\mathcal{A}$ and $\mathcal{I}$.

The first step is to create a unitary operator that maps $|i\rangle|\bar{0}\rangle \mapsto |i\rangle|R_i\rangle$ such that, for all $i \in \mathcal{I}$, after measuring the state $|R_i\rangle$, with probability at least $1 - \delta_0$, where $\delta_0 = O\big(\delta^2/(T^2|\mathcal{I}|\log(T/\delta))\big)$, the outcome r_i of the first register satisfies

$$|r_i - \Lambda_i^{\text{join}}(\mathcal{A})| \leqslant \epsilon_0. \tag{14}$$

For such, we use Lemma 13 to build maps $|i\rangle|\bar{0}\rangle \mapsto |i\rangle|\Psi_i\rangle$ and $|i\rangle|\bar{0}\rangle \mapsto |i\rangle|\Phi_i\rangle$ whose respective outcomes ψ_i and ϕ_i of the first registers satisfy

$$\begin{aligned} |\psi_i - \mathbf{X}_i^{\top}(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\mu})| &\leqslant \epsilon_1 \|\mathbf{X}_i\|_\infty \|\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\mu}\|_1 \leqslant \epsilon_1 \|\mathbf{X}\|_{\max} \|\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\mu}\|_1, \\ |\phi_i - \mathbf{X}_i^{\top}\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}| &\leqslant \epsilon_2 \|\mathbf{X}_i\|_\infty \|\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}\|_1 \leqslant \epsilon_2 \|\mathbf{X}\|_{\max} \|\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}\|_1, \end{aligned}$$

in time $O(\epsilon_1^{-1}\log(1/\delta_0)\operatorname{poly}\log(nd))$ and $O(\epsilon_2^{-1}\log(1/\delta_0)\operatorname{poly}\log(nd))$, respectively. Then, by using Lemma 7, we can construct the desired map $|i\rangle|\bar{0}\rangle \mapsto |i\rangle|R_i\rangle$ as in Equation (14) with

$$\epsilon_0 = \max_{i \in \mathcal{I}} \left\{ \Lambda_i^{\text{join}}(\mathcal{A}) \left(\frac{\epsilon_1 \|\mathbf{X}\|_{\max} \|\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\mu}\|_1}{|\mathbf{X}_i^{\top}(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\mu})|} + \frac{\epsilon_2 \|\mathbf{X}\|_{\max} \|\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}\|_1}{|\pm 1 - \mathbf{X}_i^{\top}\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}|} \right) \right\}. \tag{15}$$

Let us upper bound ϵ_0. First note that $\max_{i\in\mathcal{I}}\{\Lambda_i^{\text{join}}(\mathcal{A})\} = \lambda_{t+1}^{\text{join}}$ by the definition of $\lambda_{t+1}^{\text{join}}$. Regarding the other quantities,

$$\begin{aligned} \|\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}\|_1 = \|(\mathbf{X}_{\mathcal{A}}^{+})^{\top}\boldsymbol{\eta}_{\mathcal{A}}\|_1 &= \frac{1}{\lambda_t}\|(\mathbf{X}_{\mathcal{A}}^{+})^{\top}\mathbf{X}_{\mathcal{A}}^{\top}(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t))\|_1 \\ &\leqslant \frac{1}{\lambda_t}\|\mathbf{X}_{\mathcal{A}}^{+}\|_\infty \|\mathbf{X}_{\mathcal{A}}^{\top}(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t))\|_\infty && \text{(Hölder's inequality)} \\ &\leqslant \|\mathbf{X}_{\mathcal{A}}^{+}\|_\infty && (\|\mathbf{X}_{\mathcal{A}}^{\top}(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t))\|_\infty \leqslant \lambda_t) \\ &\leqslant \sqrt{n}\|\mathbf{X}_{\mathcal{A}}^{+}\|_2, && (\|\mathbf{X}_{\mathcal{A}}^{+}\|_\infty \leqslant \sqrt{n}\|\mathbf{X}_{\mathcal{A}}^{+}\|_2) \end{aligned}$$

using that $\|\mathbf{X}_{\mathcal{A}}^{\top}(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t))\|_\infty \leqslant \lambda_t$ since $\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t)$ satisfies the $\mathsf{KKT}_{\lambda_t}(\epsilon)$ condition. Also,

$$\begin{aligned} |\pm 1 - \mathbf{X}_i^{\top}\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}| &\geqslant 1 - |\mathbf{X}_i^{\top}\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}| \\ &= 1 - \frac{1}{\lambda_t}|\mathbf{X}_i^{\top}(\mathbf{X}_{\mathcal{A}}^{+})^{\top}\mathbf{X}_{\mathcal{A}}^{\top}(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t))| \\ &\geqslant 1 - \frac{1}{\lambda_t}\|\mathbf{X}_{\mathcal{A}}^{+}\mathbf{X}_i\|_1 \|\mathbf{X}_{\mathcal{A}}^{\top}(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t))\|_\infty && \text{(Hölder's inequality)} \\ &\geqslant 1 - \alpha_{\mathcal{A}}, && (\|\mathbf{X}_{\mathcal{A}}^{\top}(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t))\|_\infty \leqslant \lambda_t \text{ and } \|\mathbf{X}_{\mathcal{A}}^{+}\mathbf{X}_i\|_1 \leqslant \alpha_{\mathcal{A}}) \end{aligned}$$

The above inequalities are sufficient to bound the second term in Equation (15). Regarding the

first term,

$$\begin{aligned}
\max_{i\in\mathcal{I}}\left\{\left|\frac{\mathbf{X}_i^\top(\mathbf{y}-\mathbf{X}_\mathcal{A}\boldsymbol{\mu})}{\pm 1-\mathbf{X}_i^\top\mathbf{X}_\mathcal{A}\boldsymbol{\theta}}\right|\frac{\epsilon_1\|\mathbf{X}\|_{\max}\|\mathbf{y}-\mathbf{X}_\mathcal{A}\boldsymbol{\mu}\|_1}{|\mathbf{X}_i^\top(\mathbf{y}-\mathbf{X}_\mathcal{A}\boldsymbol{\mu})|}\right\} &= \epsilon_1\max_{i\in\mathcal{I}}\left\{\frac{\|\mathbf{X}\|_{\max}\|\mathbf{y}-\mathbf{X}_\mathcal{A}\boldsymbol{\mu}\|_1}{|\pm 1-\mathbf{X}_i^\top\mathbf{X}_\mathcal{A}\boldsymbol{\theta}|}\right\}\\
&\leqslant \frac{\epsilon_1}{1-\alpha_\mathcal{A}}\|\mathbf{X}\|_{\max}\|\mathbf{y}-\mathbf{X}_\mathcal{A}\boldsymbol{\mu}\|_1\\
&= \frac{\epsilon_1}{1-\alpha_\mathcal{A}}\|\mathbf{X}\|_{\max}\|\mathbf{y}-\mathbf{X}_\mathcal{A}\boldsymbol{\mu}\|_1\frac{\lambda_{t+1}^{\text{join}}}{\max_{i\in\mathcal{I}}\left|\frac{\mathbf{X}_i^\top(\mathbf{y}-\mathbf{X}_\mathcal{A}\boldsymbol{\mu})}{\pm 1-\mathbf{X}_i^\top\mathbf{X}_\mathcal{A}\boldsymbol{\theta}}\right|}\\
&\leqslant \frac{(1+\alpha_\mathcal{A})\epsilon_1\lambda_{t+1}^{\text{join}}}{1-\alpha_\mathcal{A}}\frac{\|\mathbf{X}\|_{\max}\|\mathbf{y}-\mathbf{X}_\mathcal{A}\boldsymbol{\mu}\|_1}{\max_{i\in\mathcal{I}}|\mathbf{X}_i^\top(\mathbf{y}-\mathbf{X}_\mathcal{A}\boldsymbol{\mu})|}\\
&\leqslant \frac{(1+\alpha_\mathcal{A})\epsilon_1\gamma_\mathcal{A}^{-1}\lambda_{t+1}^{\text{join}}}{1-\alpha_\mathcal{A}},
\end{aligned}$$

using that $\max_{i\in\mathcal{I}}|\mathbf{X}_i^\top(\mathbf{y}-\mathbf{X}_\mathcal{A}\boldsymbol{\mu})| = \|\mathbf{X}_\mathcal{I}^\top(\mathbf{y}-\mathbf{X}_\mathcal{A}\boldsymbol{\mu})\|_\infty = \|\mathbf{X}^\top(\mathbf{y}-\mathbf{X}_\mathcal{A}\boldsymbol{\mu})\|_\infty \geqslant \gamma_\mathcal{A}\|\mathbf{X}\|_{\max}\|\mathbf{y}-\mathbf{X}_\mathcal{A}\boldsymbol{\mu}\|_1$ and that $1-\alpha_\mathcal{A} \leqslant |\pm 1-\mathbf{X}_i^\top\mathbf{X}_\mathcal{A}\boldsymbol{\theta}| \leqslant 1+\alpha_\mathcal{A}$. All the above leads to

$$\epsilon_0 \leqslant \frac{\epsilon_1(1+\alpha_\mathcal{A})\gamma_\mathcal{A}^{-1}+\epsilon_2\sqrt{n}\|\mathbf{X}\|_{\max}\|\mathbf{X}_\mathcal{A}^+\|_2}{1-\alpha_\mathcal{A}}\lambda_{t+1}^{\text{join}}.$$

By taking

$$\epsilon_1 = \frac{\epsilon(1-\alpha_\mathcal{A})\gamma_\mathcal{A}}{2(1+\alpha_\mathcal{A})^2} \quad\text{and}\quad \epsilon_2 = \frac{\epsilon(1-\alpha_\mathcal{A})}{2(1+\alpha_\mathcal{A})\sqrt{n}\|\mathbf{X}\|_{\max}\|\mathbf{X}_\mathcal{A}^+\|_2},$$

then $\epsilon_0 \leqslant \frac{\epsilon}{2(1+\alpha_\mathcal{A})}\lambda_{t+1}^{\text{join}}$ in time $O\Big(\frac{\gamma_\mathcal{A}^{-1}+\sqrt{n}\|\mathbf{X}\|_{\max}\|\mathbf{X}_\mathcal{A}^+\|_2}{(1-\alpha_\mathcal{A})\epsilon}\log(1/\delta_0)\operatorname{poly}\log(nd)\Big)$. Finally, we apply the approximate quantum minimum-finding algorithm (Fact 11) to obtain an index $\widetilde{i}_{t+1}^{\text{join}} \in \mathcal{I}$ such that

$$\Lambda_{\widetilde{i}_{t+1}^{\text{join}}}^{\text{join}}(\mathcal{A}) \geqslant \max_{i\in\mathcal{I}}\{\Lambda_i^{\text{join}}(\mathcal{A})\} - 2\epsilon_0 \geqslant \Big(1-\frac{\epsilon}{1+\alpha_\mathcal{A}}\Big)\max_{i\in\mathcal{I}}\{\Lambda_i^{\text{join}}(\mathcal{A})\},$$

i.e., $\widetilde{i}_{t+1}^{\text{join}}$ belongs to the set from Equation (13) with probability at least $1-\delta/T$ as promised. The total complexity is $\widetilde{O}(\sqrt{|\mathcal{I}|}\log(T/\delta))$ times the complexity of constructing the mapping $|i\rangle|\bar{0}\rangle \mapsto |i\rangle|R_i\rangle$. The final success probability follows from a union bound. □

If $\|\mathbf{X}\|_{\max}\|\mathbf{X}_\mathcal{A}^+\|_2$ is constant and the parameters $\alpha_\mathcal{A}, \gamma_\mathcal{A} \in (0,1)$ are bounded away from 1 and 0, respectively, then the complexity of Algorithm 3 is $\widetilde{O}(\sqrt{n|\mathcal{I}|}/\epsilon + n|\mathcal{A}| + |\mathcal{A}|^2)$ per iteration. If also $|\mathcal{A}| = O(n)$, then we obtain a fully quadratic advantage $\widetilde{O}(\sqrt{nd})$ over the classical LARS algorithm.

4.3 Approximate classical LARS algorithm

By using similar techniques and results behind the approximate quantum LARS algorithm, it is possible to devise an analogous approximate classical LARS algorithm, which can be seen as a dequantised counterpart to our quantum algorithms. The idea is again to approximately compute the joining times $\Lambda_i^{\text{join}}(\mathcal{A}) = \frac{\mathbf{X}_i^\top(\mathbf{y}-\mathbf{X}_\mathcal{A}\boldsymbol{\mu})}{\pm 1-\mathbf{X}_i^\top\mathbf{X}_\mathcal{A}\boldsymbol{\theta}}$, but by using the classical sampling procedure from Lemma 14 instead. For well-behaved design matrices, the complexity per iteration is $\widetilde{O}(n|\mathcal{I}|/\epsilon^2 + n|\mathcal{A}| + |\mathcal{A}|^2)$. The approximate classical LARS algorithm for the pathwise Lasso is shown in Algorithm 4.

Algorithm 4: Approximate classical LARS algorithm for the pathwise Lasso

Input: $T \in \mathbb{N}$, $\delta \in (0,1)$, $\epsilon > 0$, $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n\times d}$, and $\{\alpha_{\mathcal{A}} \geqslant \|\mathbf{X}_{\mathcal{A}}^{+}\mathbf{X}_{\mathcal{A}^c}\|_1\}_{\mathcal{A}\subseteq[d]:|\mathcal{A}|\leqslant T}$

Output: Regularisation path $\widetilde{\mathcal{P}}$ with T kinks and error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}(\lambda)\|_1$ with probability $\geqslant 1-\delta$

Initialise $\mathcal{A} = \operatorname{argmax}_{j\in[d]} |\mathbf{X}_j^\top \mathbf{y}|$, $\mathcal{I} = [d]\backslash\mathcal{A}$, $\lambda_0 = \|\mathbf{X}^\top\mathbf{y}\|_\infty$, $\widetilde{\boldsymbol{\beta}}(\lambda_0) = \mathbf{0}$, $t = 0$

while $\mathcal{I} \neq \varnothing$, $t \leqslant T$ **do**

  $\boldsymbol{\eta}_{\mathcal{A}} \leftarrow \frac{1}{\lambda_t}\mathbf{X}_{\mathcal{A}}^\top(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_t))$

  $\boldsymbol{\mu} \leftarrow \mathbf{X}_{\mathcal{A}}^{+}\mathbf{y}$ // $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{A}|}$

  $\boldsymbol{\theta} \leftarrow (\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{+}\boldsymbol{\eta}_{\mathcal{A}}$ // $\boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{A}|}$

  Input $\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\mu}$ and $\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}$ into classical-samplable memories

  Define $\Lambda_i^{\text{join}} \triangleq \frac{\mathbf{X}_i^\top(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\boldsymbol{\mu})}{\pm 1-\mathbf{X}_i^\top\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}}$ and $\Lambda_i^{\text{cross}} \triangleq \frac{\mu_i}{\theta_i}\cdot\mathbf{1}\left[\frac{\mu_i}{\theta_i} \leqslant \lambda_t\right]$

  $i_{t+1}^{\text{cross}} \leftarrow \operatorname{argmax}_{i\in\mathcal{A}}\{\Lambda_i^{\text{cross}}\}$ and $\lambda_{t+1}^{\text{cross}} \leftarrow \max_{i\in\mathcal{A}}\{\Lambda_i^{\text{cross}}\}$

  Compute $\{r_i\}_{i\in\mathcal{I}}$ such that $|r_i - \Lambda_i^{\text{join}}| \leqslant \frac{\epsilon\lambda_{t+1}^{\text{join}}}{1+\alpha_{\mathcal{A}}}$ with failure probability $\frac{\delta}{T|\mathcal{I}|}$ (Lemma 14)

  $\widetilde{i}_{t+1}^{\text{join}} \leftarrow \operatorname{argmax}_{i\in\mathcal{I}}\{r_i\}$ and $\widetilde{\lambda}_{t+1}^{\text{join}} \leftarrow \left(1-\frac{\epsilon}{1+\alpha_{\mathcal{A}}}\right)^{-1}\max_{i\in\mathcal{I}}\{r_i\}$

  $\lambda_{t+1} \leftarrow \min\{\lambda_t, \max\{\lambda_{t+1}^{\text{cross}}, \widetilde{\lambda}_{t+1}^{\text{join}}\}\}$

  $\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_{t+1}) \leftarrow \boldsymbol{\mu} - \lambda_{t+1}\boldsymbol{\theta}$

  if $\lambda_{t+1} = \lambda_{t+1}^{\text{cross}}$ **then**

    Move i_{t+1}^{cross} from $\mathcal{A}$ to $\mathcal{I}$

  else

    Move $\widetilde{i}_{t+1}^{\text{join}}$ from $\mathcal{I}$ to $\mathcal{A}$

  $t \leftarrow t+1$

return coefficients $[(\lambda_0, \widetilde{\boldsymbol{\beta}}(\lambda_0)), (\lambda_1, \widetilde{\boldsymbol{\beta}}(\lambda_1)), \dots]$

Theorem 28. *Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{y} \in \mathbb{R}^n$. Assume access to classical-samplable structures of size $O(n)$. Let $\delta \in (0,1)$, $\epsilon > 0$, and $T \in \mathbb{N}$. For $\mathcal{A} \subseteq [d]$ of size $|\mathcal{A}| \leqslant T$, assume there are $\alpha_{\mathcal{A}}, \gamma_{\mathcal{A}} \in (0,1)$ such that*

$$\|\mathbf{X}_{\mathcal{A}}^{+}\mathbf{X}_{\mathcal{A}^c}\|_1 \leqslant \alpha_{\mathcal{A}} \quad \text{and} \quad \frac{\|\mathbf{X}^\top(\mathbf{I}-\mathbf{X}_{\mathcal{A}}\mathbf{X}_{\mathcal{A}}^{+})\mathbf{y}\|_\infty}{\|\mathbf{X}\|_{\max}\|(\mathbf{I}-\mathbf{X}_{\mathcal{A}}\mathbf{X}_{\mathcal{A}}^{+})\mathbf{y}\|_1} \geqslant \gamma_{\mathcal{A}}.$$

The approximate classical LARS Algorithm 4 *outputs, with probability at least $1-\delta$, a continuous piecewise linear approximate regularisation path $\widetilde{\mathcal{P}} = \{\widetilde{\boldsymbol{\beta}}(\lambda) \in \mathbb{R}^d : \lambda > 0\}$ with error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}(\lambda)\|_1$ and at most T kinks in time*

$$O\left(\frac{\gamma_{\mathcal{A}}^{-2} + n\|\mathbf{X}\|_{\max}^2\|\mathbf{X}_{\mathcal{A}}^{+}\|_2^2}{(1-\alpha_{\mathcal{A}})^2\epsilon^2}|\mathcal{I}|\log(Td/\delta)\operatorname{poly}\log n + n|\mathcal{A}| + |\mathcal{A}|^2\right)$$

per iteration, where $\mathcal{A}$ and $\mathcal{I}$ are the active and inactive sets of the corresponding iteration.

Proof. We start by noticing that the joining time computation step outputs $\widetilde{i}_{t+1}^{\text{join}} = \operatorname{argmax}_{i\in\mathcal{I}}\{r_i\}$ and $\widetilde{\lambda}_{t+1}^{\text{join}} = \left(1-\frac{\epsilon}{1+\alpha_{\mathcal{A}}}\right)^{-1}\max_{i\in\mathcal{I}}\{r_i\}$, where $|r_i - \Lambda_i^{\text{join}}(\mathcal{A})| \leqslant \frac{\epsilon}{1+\alpha_{\mathcal{A}}}\lambda_{t+1}^{\text{join}}$. Since $\max_{i\in\mathcal{I}}\{r_i\} \geqslant r_{i_{t+1}^{\text{join}}} \geqslant \Lambda_{i_{t+1}^{\text{join}}}^{\text{join}}(\mathcal{A}) - \frac{\epsilon}{1+\alpha_{\mathcal{A}}}\lambda_{t+1}^{\text{join}} = \left(1-\frac{\epsilon}{1+\alpha_{\mathcal{A}}}\right)\lambda_{t+1}^{\text{join}}$, this means the joining variable $\widetilde{i}_{t+1}^{\text{join}}$ is taken from the set

$$\left\{j \in \mathcal{I} : \Lambda_j^{\text{join}}(\mathcal{A}) \geqslant \left(1-\frac{\epsilon}{1+\alpha_{\mathcal{A}}}\right)\max_{i\in\mathcal{I}}\{\Lambda_i^{\text{join}}(\mathcal{A})\}\right\}.$$

Theorem 26 thus guarantees the correctness of the algorithm. Regarding the time complexity, the most expensive steps are computing $\mathbf{X}_{\mathcal{A}}^{+}$ in time $O(n|\mathcal{A}| + |\mathcal{A}|^2)$, inputting $\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\mu}$ and $\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}$ into classical-samplable memories in time $O(n \log n)$, and computing the $|\mathcal{I}|$ values $\{r_i\}_{i\in\mathcal{I}}$. By following the exact same steps as in the proof of Theorem 27, approximating any quantity $\Lambda_i^{\text{join}}(\mathcal{A})$ within precision $\frac{\epsilon}{1+\alpha_{\mathcal{A}}}\lambda_{t+1}^{\text{join}}$ and failure probability at most $\frac{\delta}{T|\mathcal{I}|}$ requires time

$$O\left(\frac{\gamma_{\mathcal{A}}^{-2} + n\|\mathbf{X}\|_{\max}^2\|\mathbf{X}_{\mathcal{A}}^{+}\|_2^2}{(1-\alpha_{\mathcal{A}})^2\epsilon^2}\log(Td/\delta)\operatorname{poly}\log n\right)$$

(note that the term $O(\operatorname{poly}\log n)$ comes from sampling the vectors $\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\mu}$ and $\mathbf{X}_{\mathcal{A}}\boldsymbol{\theta}$). The final success probability follows from a union bound. □

If $\|\mathbf{X}\|_{\max}\|\mathbf{X}_{\mathcal{A}}^{+}\|_2 = o(1)$ and the parameters $\alpha_{\mathcal{A}}$ and $\gamma_{\mathcal{A}}$ are bounded away from 1 and 0, respectively, then it is possible to obtain a speedup compared to the standard LARS algorithm as we shall see in the next section. For several inputs $\mathbf{X}$ and $\mathbf{y}$, however, we expect Algorithm 4 to perform worse than Algorithm 1. We interpret this as evidence for the difficulty in properly dequantising the approximate quantum LARS Algorithm 3. For ill-behaved design matrices, approximating a function of its entries should be as costly as exactly computing it.

4.4 Gaussian random design matrix

For $\mathcal{A} \subseteq [d]$, the complexity of the approximate LARS algorithms depends on the quantities

$$\|\mathbf{X}\|_{\max}\|\mathbf{X}_{\mathcal{A}}^{+}\|_2, \qquad \|\mathbf{X}_{\mathcal{A}}^{+}\mathbf{X}_{\mathcal{A}^c}\|_1, \qquad \text{and} \qquad \frac{\|\mathbf{X}^\top(\mathbf{I} - \mathbf{X}_{\mathcal{A}}\mathbf{X}_{\mathcal{A}}^{+})\mathbf{y}\|_\infty}{\|\mathbf{X}\|_{\max}\|(\mathbf{I} - \mathbf{X}_{\mathcal{A}}\mathbf{X}_{\mathcal{A}}^{+})\mathbf{y}\|_1}.$$

In this section, we shall bound these quantities for the specific case when the design matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ is a random matrix from the $\boldsymbol{\Sigma}$-Gaussian ensemble. According to Fact 19, in this situation the Lasso solution is unique almost surely, thus $\operatorname{null}(\mathbf{X}_{\mathcal{A}}) = \{\mathbf{0}\}$ and $\mathbf{X}_{\mathcal{A}}^{+} = (\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{X}_{\mathcal{A}}^\top$. As we shall prove below, with high probability, $\|\mathbf{X}\|_{\max}\|\mathbf{X}_{\mathcal{A}}^{+}\|_2$ is $O(\sqrt{\log(d)/n})$ and the mutual incoherence is at most $1/2$. The mutual overlap between $\mathbf{y}$ and $\mathbf{X}$, on the other hand, is $\Omega(\|\mathbf{y}\|_2/(\|\mathbf{y}\|_1\sqrt{\log d}))$ with high probability, which equals $\Omega(1/\sqrt{\log d})$ when $\mathbf{y}$ is sparse. These bounds yield quantum and classical complexities $\widetilde{O}(\sqrt{d}/\epsilon)$ and $\widetilde{O}(d/\epsilon^2)$ per iteration for Algorithms 3 and 4, respectively, as stated in Result 3.

Let us start our analysis with the quantity $\|\mathbf{X}\|_{\max}\|\mathbf{X}_{\mathcal{A}}^{+}\|_2$. The next result states that, if $\boldsymbol{\Sigma}$ is well behaved, which includes $\boldsymbol{\Sigma} = \mathbf{I}$, then $\|\mathbf{X}\|_{\max}\|\mathbf{X}_{\mathcal{A}}^{+}\|_2 = O(\sqrt{\log(d)/n})$ with high probability.

Lemma 29. *Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be a random matrix from the $\boldsymbol{\Sigma}$-Gaussian ensemble, $\boldsymbol{\Sigma} \succ \mathbf{0}$. Given $\mathcal{A} \subseteq [d]$ of size $|\mathcal{A}| \leqslant n$, let $\boldsymbol{\Sigma}_{\mathcal{A}|\mathcal{A}^c} = \boldsymbol{\Sigma}_{\mathcal{A}\mathcal{A}} - \boldsymbol{\Sigma}_{\mathcal{A}\mathcal{A}^c}(\boldsymbol{\Sigma}_{\mathcal{A}^c\mathcal{A}^c})^{-1}\boldsymbol{\Sigma}_{\mathcal{A}^c\mathcal{A}} \succ \mathbf{0}$ be the conditional covariance matrix of $\mathbf{X}_{\mathcal{A}}|\mathbf{X}_{\mathcal{A}^c}$. Then, for $\delta \in (2e^{-n/2}, 1)$,*

$$\mathbb{P}\left[\|\mathbf{X}\|_{\max}\|\mathbf{X}_{\mathcal{A}}^{+}\|_2 \leqslant \frac{\|\sqrt{\boldsymbol{\Sigma}}\|_1\sqrt{\ln(4nd/\delta)}}{(\sqrt{n} - \sqrt{2\ln(2/\delta)})\sigma_{\min}(\sqrt{\boldsymbol{\Sigma}_{\mathcal{A}|\mathcal{A}^c}}) - \sqrt{\operatorname{Tr}[\boldsymbol{\Sigma}_{\mathcal{A}|\mathcal{A}^c}]}}\right] \geqslant 1 - \delta.$$

In particular, if $\boldsymbol{\Sigma} = \mathbf{I}$ and $|\mathcal{A}| \leqslant n/2$, then

$$\mathbb{P}\left[\|\mathbf{X}\|_{\max}\|\mathbf{X}_{\mathcal{A}}^{+}\|_2 \leqslant \frac{\sqrt{\ln(4nd/\delta)}}{(1-1/\sqrt{2})\sqrt{n} - \sqrt{2\ln(2/\delta)}}\right] \geqslant 1 - \delta.$$

Proof. Let us start by bounding $\|\mathbf{X}\|_{\max}$. We can write $\mathbf{X} = \mathbf{Z}\sqrt{\mathbf{\Sigma}}$ where $\mathbf{Z}$ is drawn from the $\mathbf{I}$-Gaussian ensemble. Let $\mathbf{Z}^{(i)} \in \mathbb{R}^{d\times 1}$ be the i-th row of $\mathbf{Z}$. Then

$$\|\mathbf{X}\|_{\max} = \|\mathbf{Z}\sqrt{\mathbf{\Sigma}}\|_{\max} = \max_{i\in[n]} \|\sqrt{\mathbf{\Sigma}}^\top \mathbf{Z}^{(i)}\|_\infty \leqslant \max_{i\in[n]} \|\sqrt{\mathbf{\Sigma}}^\top\|_\infty \|\mathbf{Z}^{(i)}\|_\infty = \|\sqrt{\mathbf{\Sigma}}\|_1 \|\mathbf{Z}\|_{\max}.$$

By a Chernoff bound (Fact 3) plus a union bound, we have that $\mathbb{P}[\|\mathbf{Z}\|_{\max} \leqslant \sqrt{\ln(4nd/\delta)}] \geqslant 1-\delta/2$, which implies that $\mathbb{P}[\|\mathbf{X}\|_{\max} \leqslant \sqrt{\ln(4nd/\delta)}\|\sqrt{\mathbf{\Sigma}}\|_1] \geqslant 1-\delta/2$.

We now focus on $\|\mathbf{X}_{\mathcal{A}}^+\|_2$. The rows of $\mathbf{X}_{\mathcal{A}}$ are i.i.d. sampled from the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathcal{A}|\mathcal{A}^c})$. According to [Wai19, Theorem 6.1],

$$\mathbb{P}\left[\|\mathbf{X}_{\mathcal{A}}^+\|_2^{-1} \leqslant (1-\epsilon)\sqrt{n}\sigma_{\min}(\sqrt{\mathbf{\Sigma}_{\mathcal{A}|\mathcal{A}^c}}) - \sqrt{\mathrm{Tr}[\mathbf{\Sigma}_{\mathcal{A}|\mathcal{A}^c}]}\right] \leqslant e^{-n\epsilon^2/2} \quad \text{for all } \epsilon \in (0,1).$$

By taking $\epsilon = \sqrt{2\ln(2/\delta)/n} < 1$, the above probability is at most $\delta/2$. The result follows by putting together the bounds on $\|\mathbf{X}\|_{\max}$ and $\|\mathbf{X}_{\mathcal{A}}^+\|_2$.

In case $\mathbf{\Sigma} = \mathbf{I}$, then $\mathbf{\Sigma}_{\mathcal{A}|\mathcal{A}^c} = \mathbf{I}$, $\sigma_{\min}(\sqrt{\mathbf{\Sigma}_{\mathcal{A}|\mathcal{A}^c}}) = 1$, $\mathrm{Tr}[\mathbf{\Sigma}_{\mathcal{A}|\mathcal{A}^c}] = |\mathcal{A}|$, and $\|\mathbf{\Sigma}\|_1 = 1$. Therefore, $\|\mathbf{X}_{\mathcal{A}}^+\|_2^{-1} \geqslant \sqrt{n} - \sqrt{|\mathcal{A}|} - \sqrt{2\ln(2/\delta)} \geqslant (1-1/\sqrt{2})\sqrt{n} - \sqrt{2\ln(2/\delta)}$ with high probability. □

We now move on to $\|\mathbf{X}_{\mathcal{A}}^+\mathbf{X}_{\mathcal{A}^c}\|_1$. The next lemma tells us that, if $|\mathcal{A}| = O(\sigma_{\min}(\mathbf{\Sigma})n/\log d)$, where $\sigma_{\min}(\mathbf{\Sigma})$ is the minimum eigenvalue of $\mathbf{\Sigma}$, then the mutual incoherence is a constant bounded away from 1 with high probability. A similar result can be found in [Wai19, Exercise 7.19].

Lemma 30. *Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be a random matrix from the $\mathbf{\Sigma}$-Gaussian ensemble, where $\mathbf{\Sigma} \succ \mathbf{0}$ has minimum eigenvalue $\sigma_{\min}(\mathbf{\Sigma}) \in (0,1]$ and its diagonal entries are at most 1. Given $\mathcal{A} \subseteq [d]$, suppose that there is $\bar{\alpha} \in [0,1)$ such that*

$$\max_{j\in\mathcal{A}^c} \|\mathbf{\Sigma}_{j\mathcal{A}}(\mathbf{\Sigma}_{\mathcal{A}\mathcal{A}})^{-1}\|_1 \leqslant \bar{\alpha}.$$

Given $\delta \in (0,1)$, if $|\mathcal{A}| \leqslant (1-\bar{\alpha})^2\sigma_{\min}(\mathbf{\Sigma})n/72\ln(4d/\delta)$, then

$$\mathbb{P}\left[\|\mathbf{X}_{\mathcal{A}}^+\mathbf{X}_{\mathcal{A}^c}\|_1 \leqslant \frac{1}{2} + \frac{\bar{\alpha}}{2}\right] \geqslant 1-\delta.$$

Proof. First notice that

$$\|\mathbf{X}_{\mathcal{A}}^+\mathbf{X}_{\mathcal{A}^c}\|_1 = \|(\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}^c}\|_1 = \|\mathbf{X}_{\mathcal{A}^c}^\top\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{-1}\|_\infty.$$

For any $\mathbf{u} \in \mathbb{R}^{|\mathcal{A}|}$ such that $\|\mathbf{u}\|_\infty = 1$, note that

$$\|\mathbf{X}_{\mathcal{A}^c}^\top\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{-1}\|_\infty \geqslant \|\mathbf{X}_{\mathcal{A}^c}^\top\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}\|_\infty = \max_{j\in\mathcal{A}^c} |\mathbf{X}_j^\top\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}|.$$

Consider the quantity $|\mathbf{X}_j^\top\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}|$ for fixed $j \in \mathcal{A}^c$. The zero-mean Gaussian vector $\mathbf{X}_j$ can be decomposed into a linear prediction based on $\mathbf{X}_{\mathcal{A}}$ plus a prediction error as (cf. [Wai09, Section V.A] and [Wai19, Equation (11.26) & Exercise 11.3])

$$\mathbf{X}_j^\top = \mathbf{\Sigma}_{j\mathcal{A}}(\mathbf{\Sigma}_{\mathcal{A}\mathcal{A}})^{-1}\mathbf{X}_{\mathcal{A}}^\top + \mathbf{W}_j^\top, \tag{16}$$

where $\mathbf{W}_j \in \mathbb{R}^n$ is a vector with i.i.d. $\mathcal{N}(0, [\mathbf{\Sigma}_{\mathcal{A}^c|\mathcal{A}}]_{jj})$ entries. Here the (positive-definite) matrix $\mathbf{\Sigma}_{\mathcal{A}^c|\mathcal{A}} = \mathbf{\Sigma}_{\mathcal{A}^c\mathcal{A}^c} - \mathbf{\Sigma}_{\mathcal{A}^c\mathcal{A}}(\mathbf{\Sigma}_{\mathcal{A}\mathcal{A}})^{-1}\mathbf{\Sigma}_{\mathcal{A}\mathcal{A}^c}$ is the conditional covariance matrix of $\mathbf{X}_{\mathcal{A}^c}|\mathbf{X}_{\mathcal{A}}$. Thus

$$\begin{aligned}
|\mathbf{X}_j^\top\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}| &\leqslant |\mathbf{\Sigma}_{j\mathcal{A}}(\mathbf{\Sigma}_{\mathcal{A}\mathcal{A}})^{-1}\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}| + |\mathbf{W}_j^\top\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}| \\
&\leqslant \|\mathbf{\Sigma}_{j\mathcal{A}}(\mathbf{\Sigma}_{\mathcal{A}\mathcal{A}})^{-1}\|_1\|\mathbf{u}\|_\infty + |\mathbf{W}_j^\top\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}| \\
&\leqslant \bar{\alpha} + |\mathbf{W}_j^\top\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}|.
\end{aligned}$$

Since $\mathbf{W}_j$ is independent of $\mathbf{X}_{\mathcal{A}}$, $\mathbf{W}_j^\top \mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}$ is a sub-Gaussian random variable with parameter (Fact 4)

$$[\boldsymbol{\Sigma}_{\mathcal{A}^c|\mathcal{A}}]_{jj}\|\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}\|_2^2 \leqslant \|\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}\|_2^2 \\ = \mathbf{u}^\top(\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u} \leqslant \|\mathbf{u}\|_2^2\|(\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}})^{-1}\|_2 \leqslant |\mathcal{A}|\|(\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}})^{-1}\|_2,$$

where we used that $[\boldsymbol{\Sigma}_{\mathcal{A}^c|\mathcal{A}}]_{jj} \leqslant 1$ since $\boldsymbol{\Sigma}_{\mathcal{A}^c|\mathcal{A}} \preceq \boldsymbol{\Sigma}_{\mathcal{A}^c\mathcal{A}^c}$ and the diagonal entries of $\boldsymbol{\Sigma}$ are at most 1. We now relate $\|(\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}})^{-1}\|_2$ with $\sigma_{\min}(\boldsymbol{\Sigma}_{\mathcal{A}\mathcal{A}}) \geqslant \sigma_{\min}(\boldsymbol{\Sigma})$. According to [Wai09, Lemma 9],

$$\mathbb{P}\left[\|(\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}})^{-1}\|_2 \geqslant \frac{(1+t+\sqrt{|\mathcal{A}|/n})^2}{n\sigma_{\min}(\boldsymbol{\Sigma})}\right] \leqslant 2e^{-nt^2/2} \quad \text{for all } t > 0.$$

Define the event $\mathcal{E}(\mathbf{X}_{\mathcal{A}}) \triangleq \{\|(\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}})^{-1}\|_2 \geqslant 9/(n\sigma_{\min}(\boldsymbol{\Sigma}))\}$. Since $|\mathcal{A}| \leqslant n$, the event $\mathcal{E}(\mathbf{X}_{\mathcal{A}})$ happens with probability at most $2e^{-n/2}$. Therefore, with probability at least $1-2e^{-n/2}$, the quantity $\mathbf{W}_j^\top \mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}$ is a sub-Gaussian random variable with parameter at most $9|\mathcal{A}|/(n\sigma_{\min}(\boldsymbol{\Sigma}))$. A Chernoff bound yields

$$\mathbb{P}\left[\max_{j\in\mathcal{A}^c}|\mathbf{W}_j^\top \mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}| \geqslant \frac{1-\bar{\alpha}}{2}\right] \leqslant \mathbb{P}\left[\max_{j\in\mathcal{A}^c}|\mathbf{W}_j^\top \mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}| \geqslant \frac{1-\bar{\alpha}}{2}\middle|\mathcal{E}^c(\mathbf{X}_{\mathcal{A}})\right] + \mathbb{P}[\mathcal{E}(\mathbf{X}_{\mathcal{A}})] \\ \leqslant 2(d-|\mathcal{A}|)\exp\left(-\frac{(1-\bar{\alpha})^2 n\sigma_{\min}(\boldsymbol{\Sigma})}{72|\mathcal{A}|}\right) + 2\exp(-n/2) \\ \leqslant 4d\exp\left(-\frac{(1-\bar{\alpha})^2 n\sigma_{\min}(\boldsymbol{\Sigma})}{72|\mathcal{A}|}\right).$$

The above probability is at most δ if $|\mathcal{A}| \leqslant (1-\bar{\alpha})^2\sigma_{\min}(\boldsymbol{\Sigma})n/72\ln(4d/\delta)$. Thus, with probability at least $1-\delta$, $|\mathbf{X}_j^\top \mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}})^{-1}\mathbf{u}| \leqslant \frac{1}{2} + \frac{\bar{\alpha}}{2}$ for any $\mathbf{u} \in \mathbb{R}^{|\mathcal{A}|}$ such that $\|\mathbf{u}\|_\infty = 1$. □

The above lemma applies to standard Gaussian matrices with $\boldsymbol{\Sigma} = \mathbf{I}$, in which case $\bar{\alpha} = 0$ and $\sigma_{\min}(\boldsymbol{\Sigma}) = 1$, and thus the mutual incoherence $\|\mathbf{X}_{\mathcal{A}}^+ \mathbf{X}_{\mathcal{A}^c}\|_1$ is less than $1/2$ with high probability if $|\mathcal{A}| = O(n/\log d)$.

Finally, we analyse the mutual overlap between $\mathbf{y}$ and $\mathbf{X}$. The next lemma states that the mutual overlap between $\mathbf{y}$ and $\mathbf{X}$ is $\Omega\Big(\frac{\|\mathbf{y}\|_2}{\|\mathbf{y}\|_1}\frac{1}{\sqrt{\log d}}\Big)$ with high probability if $|\mathcal{A}| = O(n)$, $\|\sqrt{\boldsymbol{\Sigma}}\|_1 = O(1)$, and the diagonal entries of the conditional covariance matrices $\boldsymbol{\Sigma}_{\mathcal{A}^c|\mathcal{A}}$ are bounded away from 0 (this condition can be relaxed to a constant fraction of the diagonal entries be bounded away from 0). Hence, if $\|\mathbf{y}\|_1/\|\mathbf{y}\|_2 = O(1)$, e.g., in the case when $\mathbf{y}$ is sparse, then the mutual overlap is $\Omega(1/\sqrt{\log d})$, independent of n!

Lemma 31. *Let $\mathbf{y} \in \mathbb{R}^n$ and $\mathbf{X} \in \mathbb{R}^{n\times d}$ a random matrix from the $\boldsymbol{\Sigma}$-Gaussian ensemble, $\boldsymbol{\Sigma} \succ \mathbf{0}$. Given $\mathcal{A} \subseteq [d]$, suppose that*

$$\boldsymbol{\Sigma}_{\mathcal{A}^c|\mathcal{A}} \succeq C_{\boldsymbol{\Sigma}}\mathbf{I} \quad \textit{for} \quad C_{\boldsymbol{\Sigma}} > 0,$$

where $\boldsymbol{\Sigma}_{\mathcal{A}^c|\mathcal{A}} = \boldsymbol{\Sigma}_{\mathcal{A}^c\mathcal{A}^c} - \boldsymbol{\Sigma}_{\mathcal{A}^c\mathcal{A}}(\boldsymbol{\Sigma}_{\mathcal{A}\mathcal{A}})^{-1}\boldsymbol{\Sigma}_{\mathcal{A}\mathcal{A}^c} \succ \mathbf{0}$ is the conditional covariance matrix of $\mathbf{X}_{\mathcal{A}^c}|\mathbf{X}_{\mathcal{A}}$. Given $\delta \in (0,1)$, if $|\mathcal{A}| \leqslant \min\{n/2, n - 16\ln(3/\delta)\}$, then

$$\mathbb{P}\left[\frac{\|\mathbf{X}^\top(\mathbf{I}-\mathbf{X}_{\mathcal{A}}\mathbf{X}_{\mathcal{A}}^+)\mathbf{y}\|_\infty}{\|\mathbf{X}\|_{\max}\|(\mathbf{I}-\mathbf{X}_{\mathcal{A}}\mathbf{X}_{\mathcal{A}}^+)\mathbf{y}\|_1} \geqslant \frac{(\delta/3)^{2/d}}{4\|\sqrt{\boldsymbol{\Sigma}}\|_1}\frac{\|\mathbf{y}\|_2}{\|\mathbf{y}\|_1}\sqrt{\frac{\pi C_{\boldsymbol{\Sigma}}}{\ln(6nd/\delta)}}\right] \geqslant 1-\delta.$$

Proof. Let us start by bounding $\|\mathbf{X}\|_{\max}$. Similarly to Lemma 29, $\|\mathbf{X}\|_{\max} \leqslant \|\sqrt{\mathbf{\Sigma}}\|_1 \|\mathbf{Z}\|_{\max}$, where $\mathbf{Z}$ is drawn from the $\mathbf{I}$-Gaussian ensemble. By a Chernoff bound (Fact 3) plus a union bound, $\mathbb{P}[\|\mathbf{Z}\|_{\max} \leqslant \sqrt{\ln(6nd/\delta)}] \geqslant 1 - \delta/3$, which implies that $\mathbb{P}[\|\mathbf{X}\|_{\max} \leqslant \sqrt{\ln(6nd/\delta)}\|\sqrt{\mathbf{\Sigma}}\|_1] \geqslant 1 - \delta/3$.

We now move on to the remaining part of the expression. First notice that

$$\|\mathbf{X}^\top(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}\|_\infty = \max_{j \in \mathcal{A}^c} |\mathbf{X}_j^\top(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}|.$$

Fix $j \in \mathcal{A}^c$. Again by Equation (16), we can write

$$\mathbf{X}_j^\top = \mathbf{\Sigma}_{j\mathcal{A}}(\mathbf{\Sigma}_{\mathcal{A}\mathcal{A}})^{-1}\mathbf{X}_\mathcal{A}^\top + \mathbf{W}_j^\top,$$

where $\mathbf{W}_j \in \mathbb{R}^n$ is a vector with i.i.d. $\mathcal{N}(0, [\mathbf{\Sigma}_{\mathcal{A}^c|\mathcal{A}}]_{jj})$ entries. Therefore,

$$|\mathbf{X}_j^\top(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}| = |\mathbf{W}_j^\top(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}|,$$

where we used that $\mathbf{X}_\mathcal{A}^\top(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+) = \mathbf{0}$. Since $\mathbf{W}_j$ is independent of $\mathbf{X}_\mathcal{A}$, then $\mathbf{W}_j^\top(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}$ is a Gaussian random variable with variance

$$[\mathbf{\Sigma}_{\mathcal{A}^c|\mathcal{A}}]_{jj}\|(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}\|_2^2 \geqslant C_\mathbf{\Sigma}\|(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}\|_2^2$$

since $[\mathbf{\Sigma}_{\mathcal{A}^c|\mathcal{A}}]_{jj} \geqslant C_\mathbf{\Sigma}$. Furthermore, since $\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+$ is a random projection, i.e., a projection onto a random $(n - |\mathcal{A}|)$-dimensional subspace uniformly distributed in the Grassmann manifold $G_{n,n-|\mathcal{A}|}$, then Lévy's lemma (Fact 6) yields[4] (see also [Ver18, Lemma 5.3.2])

$$\mathbb{P}\left[\|(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}\|_2 \geqslant \frac{1}{2}\sqrt{\frac{n - |\mathcal{A}|}{n}}\|\mathbf{y}\|_2\right] \geqslant 1 - e^{-(n-|\mathcal{A}|)/16}.$$

Define the event $\mathcal{E}(\mathcal{A}) \triangleq \left\{\|(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}\|_2 \geqslant \sqrt{1 - |\mathcal{A}|/n}\|\mathbf{y}\|_2/2\right\}$ and let $\operatorname{erf}(t) = \frac{2}{\sqrt{\pi}}\int_0^t e^{-z^2}\mathrm{d}z$, $t \geqslant 0$, be the error function. Then, for $t \in [0,1]$ and conditioned on the event $\mathcal{E}(\mathcal{A})$ happening,

$$\mathbb{P}\left[\frac{|\mathbf{W}_j^\top(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}|}{\|(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}\|_1} \leqslant t\sqrt{\frac{\pi C_\mathbf{\Sigma}(n - |\mathcal{A}|)}{8n}}\frac{\|\mathbf{y}\|_2}{\|\mathbf{y}\|_1} \;\middle|\; \mathcal{E}(\mathcal{A})\right] \leqslant \operatorname{erf}\left(\frac{t\sqrt{\pi}\|(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}\|_1}{2\|\mathbf{y}\|_1}\right) \leqslant t,$$

using that $\|(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}\|_1 \leqslant \|\mathbf{y}\|_1$ and $\operatorname{erf}(t) \leqslant 2t/\sqrt{\pi}$. By the independence of all $\mathbf{W}_j$, $j \in \mathcal{A}^c$,

$$\mathbb{P}\left[\max_{j \in \mathcal{A}^c}\frac{|\mathbf{W}_j^\top(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}|}{\|(\mathbf{I} - \mathbf{X}_\mathcal{A}\mathbf{X}_\mathcal{A}^+)\mathbf{y}\|_1} \leqslant t\sqrt{\frac{\pi C_\mathbf{\Sigma}(n - |\mathcal{A}|)}{8n}}\frac{\|\mathbf{y}\|_2}{\|\mathbf{y}\|_1}\right] \leqslant t^{d-|\mathcal{A}|} + e^{-(n-|\mathcal{A}|)/16}. \tag{17}$$

By taking $t = (\delta/3)^{1/(d-n)}$ and $|\mathcal{A}| \leqslant n - 16\ln(3/\delta)$, then the above probability is at most $2\delta/3$. Finally, we put Equation (17) with $n - |\mathcal{A}| \geqslant n/2$ together with the bound on $\|\mathbf{X}\|_{\max}$. □

Remark 32. *The results from this section apply to a fixed active set $\mathcal{A}$. It is possible to bound $\|\mathbf{X}\|_{\max}\|\mathbf{X}_\mathcal{A}^+\|_2$, the mutual incoherence $\alpha_\mathcal{A}$, and the mutual overlap $\gamma_\mathcal{A}$ for several sets $\mathcal{A}$ by taking a union bound on the failure probability in Lemma 29 to 31. Since the approximate regularisation path $\widetilde{\mathcal{P}}$ has at most $O(n)$ kinks for when $\mathbf{X}$ is a random Gaussian matrix (Fact 19), Algorithms 3 and 4 need only to consider $O(n)$ different active sets. Therefore, we only need to consider a union bound over $O(n)$ different sets.*

[4]The function $\mathbf{u} \in \mathbb{R}^n \mapsto \|\mathbf{P}\mathbf{u}\|_2$ is 1-Lipschitz for any orthogonal projector $\mathbf{P} \in \mathbb{R}^{n\times n}$.

5 Query lower bounds for Lasso

In this section we prove classical and quantum query lower bounds for Lasso. Our results follow from the fact that an algorithm for Lasso can be used to solve specific set-finding problems, which were studied by Chen and de Wolf [CdW23] (and also employed to derive their lower bounds).

5.1 Set-finding problems

We start by defining the set-finding problems of interest, which basically hide a set W in the columns of a matrix $\mathbf{X}$.

Definition 33 (Exact and approximate set-finding problems $\mathsf{ESF}^{d,N}_{w,p}$ and $\mathsf{ASF}^{d,N}_{w,p}$)**.** *Let $p \in (0,1/2)$ and $W \subset [d]$ of size w. Consider a matrix $\mathbf{X} \in \{-1,1\}^{N\times d}$ such that*

$$\sum_{i=1}^{N} X_{ij} = 2pN \quad \text{if } j \in W \qquad \text{and} \qquad \sum_{i=1}^{N} X_{ij} = 0 \quad \text{if } j \notin W.$$

In the exact set-finding problem $\mathsf{ESF}^{d,N}_{w,p}$ *the task is to find W. In the* approximate set-finding problem $\mathsf{ASF}^{d,N}_{w,p}$ *the task is to find $\widetilde{W} \subseteq [d]$ such that $|W \Delta \widetilde{W}| \leqslant w/5$.*

Chen and de Wolf [CdW23] proved the following query lower bound on any algorithms that solve $\mathsf{ESF}^{d,N}_{w,p}$ and $\mathsf{ASF}^{d,N}_{w,p}$.[5]

Fact 34 ([CdW23, Corollary 4.9])**.** *Let $N \in \mathbb{N}$ be even and $p \in (0,1/2)$ an integer multiple of $1/N$. Every bounded-error quantum algorithm that solves the approximate set-finding problem $\mathsf{ASF}^{d,N}_{w,p}$ with matrix $\mathbf{X} \in \{-1,1\}^{N\times d}$ uses $\Omega(\sqrt{wd}/p)$ quantum queries to $\mathbf{X}$.*

Fact 35 ([CdW23, Theorem B.5])**.** *Let $N \in \mathbb{N}$ be even, $p \in (1/\sqrt{N}, 1/2)$ an integer multiple of $1/N$, and $w = 1/2p$. Every classical algorithm that solves the exact set-finding problem $\mathsf{ESF}^{d,N}_{w,p}$ with matrix $\mathbf{X} \in \{-1,1\}^{N\times d}$ uses $\Omega((d-w)/p^2)$ queries to $\mathbf{X}$.*

We will also need a distributional version of the approximate set-finding problem.

Definition 36 (Distributional set-finding problem $\mathsf{DSF}^{d,n}_{w,p}$)**.** *Let $p \in (0,1/2)$, $n \in \mathbb{N}$, and $W \subset [d]$ of size w. Consider the distribution $\mathcal{D}_{p,W}$ over $(\mathbf{x}, y) \in \{-1,1\}^d \times \{-1,1\}$ defined as follows:*

$$\Pr[x_j = 1] = \begin{cases} \frac{1}{2} & \text{if } j \notin W, \\ \frac{1}{2} + p & \text{if } j \in W, \end{cases} \qquad \text{and} \qquad \Pr[y = 1] = 1.$$

In the distributional set-finding problem $\mathsf{DSF}^{d,n}_{w,p}$ *the task is to find $\widetilde{W} \subset [d]$ such that $|\widetilde{W} \Delta W| \leqslant w/5$ given n samples from $\mathcal{D}_{p,W}$.*

Chen and de Wolf proved that one can convert an instance of $\mathsf{ASF}^{d,N}_{w,p}$ (to be viewed as a worst-case instance) to an instance of $\mathsf{DSF}^{d,n}_{w,p}$ (to be viewed as an average-case instance).

Fact 37 ([CdW23, Theorem 4.10])**.** *Let $N \in \mathbb{N}$ be even, $n \in \mathbb{N}$, $p \in (0,1/2)$ an integer multiple of $1/n$, and $w \in (2, d/2)$ a natural number. Let $\mathbf{X} \in \{-1,1\}^{N\times d}$ be a valid input to $\mathsf{ASF}^{d,N}_{w,p}$ and let $W \subset [d]$ be the set of indices of columns of $\mathbf{X}$ whose entries add to $2pn$. Let $\mathbf{R} \in [n]^{n\times d}$ be a matrix whose entries are i.i.d. samples from the uniform distribution over $[n]$. Define $\mathbf{Z} \in \{-1,1\}^{n\times d}$ as $Z_{ij} = X_{R_{ij},j}$. Then the n vectors $(\mathbf{Z}_{i\cdot}, 1) \in \{-1,1\}^{d+1}$, where $\mathbf{Z}_{i\cdot}$ is the i-th row of $\mathbf{Z}$ and $i \in [n]$, are i.i.d. samples from the distribution $\mathcal{D}_{p,W}$ defining $\mathsf{DSF}^{d,n}_{w,p}$.*

[5] In [CdW23], Chen and de Wolf define the approximate set-finding problem as *worst-case set-finding problem* $\mathsf{WSF}_{d,w,p,N}$. We decided to change the name slightly in order to draw a clearer parallel with $\mathsf{ESF}^{d,n}_{w,p}$.

Finally, the authors also proved that an algorithm for $\mathsf{DSF}_{w,p}^{d,n}$ can be used to solve $\mathsf{ESF}_{w,p}^{d,N}$.

Fact 38 ([CdW23, Theorem B.6])**.** *Let $N \in \mathbb{N}$ be even, $p \in (1/\sqrt{N}, 1/4)$ be an integer multiple of $1/N$, and $w = 1/2p$. Suppose $\mathscr{A}$ is an algorithm for $\mathsf{DSF}_{w,p}^{d,n}$ that outputs $\widetilde{W} \subset [d]$ for every $W \subset [d]$ of size w such that $|\widetilde{W} \Delta W| \leqslant w/200$ with probability at least $1 - c/\log d$ for a small enough constant $c > 0$. Then there is an algorithm that solves $\mathsf{ESF}_{w,p}^{d,N}$ with success probability at least $99/100$ by querying $\mathscr{A}$ a number of $O(\log d)$ times.*

5.2 Solving set-finding problems with a Lasso solver

We shall now prove that an algorithm for Lasso can be used to find a good approximation to the hidden set W, after which the sought-out lower bounds will follow from Fact 34 and 35. In order to do so, we must first consider an expected version of the Lasso problem defined as

$$\hat{\boldsymbol{\beta}} \in \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^d} \overline{\mathcal{L}}_{\mathcal{D}}^{(\lambda)}(\boldsymbol{\beta}) \quad \text{where} \quad \overline{\mathcal{L}}_{\mathcal{D}}^{(\lambda)}(\boldsymbol{\beta}) \triangleq \frac{1}{2} \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} [(\mathbf{x}^\top \boldsymbol{\beta} - y)^2] + \lambda \|\boldsymbol{\beta}\|_1 \quad \text{for } \lambda > 0.$$

Here, $\mathcal{D}$ is a distribution over $(\mathbf{x}, y) \in \mathbb{R}^d \times \mathbb{R}$. We start by studying the expected Lasso $\overline{\mathcal{L}}_{\mathcal{D}_{p,W}}^{(\lambda)}(\boldsymbol{\beta})$ relative to the distribution $\mathcal{D}_{p,W}$ from Definition 36.

Lemma 39. *Let $p \in (0, 1/2)$ and $W \subset [d]$ of size w. Consider the distribution $\mathcal{D}_{p,W}$ over $\{-1,1\}^d \times \{-1,1\}$ from Definition 36. Then the unique minimiser of the expected Lasso $\overline{\mathcal{L}}_{\mathcal{D}_{p,W}}^{(\lambda)}(\boldsymbol{\beta})$ is the vector $\hat{\boldsymbol{\beta}} \in \mathbb{R}^d$ given by*

$$\hat{\beta}_j = \begin{cases} \frac{2p}{1+\lambda+4p^2(w-1)} & \text{if } j \in W, \\ 0 & \text{if } j \notin W. \end{cases}$$

Moreover, $\nabla^2 \overline{\mathcal{L}}_{\mathcal{D}_{p,W}}^{(\lambda)}(\boldsymbol{\beta}) \geq (1 - 4p^2)\mathbf{I}_d$ for all $\boldsymbol{\beta} \in \mathbb{R}^d$, $\nabla \overline{\mathcal{L}}_{\mathcal{D}_{p,W}}^{(\lambda)}(\hat{\boldsymbol{\beta}}) = \mathbf{0}$, and

$$\overline{\mathcal{L}}_{\mathcal{D}_{p,W}}^{(\lambda)}(\hat{\boldsymbol{\beta}}) = \frac{1}{2} + \frac{2pw(\lambda - 2p)}{1 + \lambda + 4p^2(w-1)} + \frac{2p^2 w(1 + 4p^2(w-1))}{(1 + \lambda + 4p^2(w-1))^2} \leqslant \frac{1}{2} + \frac{2pw(\lambda - p)}{1 + \lambda + 4p^2(w-1)}.$$

Proof. First notice that, for $i \neq j$, $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{p,W}}[x_i x_j] = 4p^2$ if $i, j \in W$ and $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{p,W}}[x_i x_j] = 0$ otherwise. Let $\hat{\boldsymbol{\beta}} \in \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \overline{\mathcal{L}}_{\mathcal{D}_{p,W}}^{(\lambda)}(\boldsymbol{\beta})$. Thus

$$\begin{aligned} \overline{\mathcal{L}}_{\mathcal{D}_{p,W}}^{(\lambda)}(\boldsymbol{\beta}) &= \frac{1}{2} \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}_{p,W}} \left[y^2 - 2y \sum_{i\in[d]} x_i \beta_i + \sum_{i,j\in[d]} x_i x_j \beta_i \beta_j \right] + \lambda \|\boldsymbol{\beta}\|_1 \\ &= \frac{1}{2} \left(1 - 4p \sum_{i\in W} \beta_i + \sum_{i\in[d]} \beta_i^2 + 4p^2 \sum_{i\neq j\in W} \beta_i \beta_j \right) + \lambda \|\boldsymbol{\beta}\|_1 \end{aligned} \tag{18}$$

and

$$\begin{aligned} \nabla \overline{\mathcal{L}}_{\mathcal{D}_{p,W}}^{(\lambda)}(\boldsymbol{\beta}) &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}_{p,W}} [(\mathbf{x}^\top \boldsymbol{\beta} - y)\mathbf{x}] + \lambda \mathbf{s} \implies \\ \nabla_j \overline{\mathcal{L}}_{\mathcal{D}_{p,W}}^{(\lambda)}(\boldsymbol{\beta}) &= \begin{cases} \beta_j + 4p^2 \sum_{i\neq j\in W} \beta_i - 2p + \lambda s_j & \text{if } j \in W, \\ \beta_j + \lambda s_j & \text{if } j \notin W, \end{cases} \end{aligned}$$

where $\mathbf{s} \in \partial\|\boldsymbol{\beta}\|_1$ is a sub-gradient. Since $\hat{\boldsymbol{\beta}}$ minimises $\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\boldsymbol{\beta})$, then $\nabla\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}}) = \mathbf{0}$. Moreover, if we arrange the indices such that $\{1,\dots,w\} = W$ and $\{w+1,\dots,d\} = [d]\backslash W$, then

$$\nabla^2\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\boldsymbol{\beta}) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}_{p,W}}[\mathbf{x}^\top\mathbf{x}] = \begin{pmatrix} 4p^2\mathbf{J}_w + (1-4p^2)\mathbf{I}_w & \mathbf{0}_{w\times\bar{w}} \\ \mathbf{0}_{\bar{w}\times w} & 2\mathbf{I}_{\bar{w}} \end{pmatrix} \geq (1-4p^2)\mathbf{I}_d,$$

where $\bar{w} \triangleq d - w$ and $\mathbf{J}_w$ is the all-ones matrix. We see that $\nabla^2\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\boldsymbol{\beta})$ is a constant matrix independent of $\boldsymbol{\beta}$. Regarding the optimal solution $\hat{\boldsymbol{\beta}}$, since $s_j = 0$ if $j \notin W$ and $s_j = \operatorname{sign}(\hat{\beta}_j)$ if $j \in W$, then

$$\begin{cases} (1-4p^2)\hat{\beta}_j + \lambda\operatorname{sign}(\hat{\beta}_j) - 2p + 4p^2\sum_{i\in W}\hat{\beta}_i = 0 & \text{if } j \in W, \\ \hat{\beta}_j = 0 & \text{if } j \notin W, \end{cases} \implies \begin{cases} \hat{\beta}_j = \frac{2p}{1+\lambda+4p^2(w-1)} & \text{if } j \in W, \\ \hat{\beta}_j = 0 & \text{if } j \notin W. \end{cases}$$

Since the Hessian $\nabla^2\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\boldsymbol{\beta})$ is positive definite, the above solution is unique. Finally, the expression for $\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}})$ follows from Equation (18) and the expression for $\hat{\boldsymbol{\beta}}$. □

We now relate the entries of an approximate solution of the expected Lasso problem to the elements of the hidden set W.

Lemma 40. *Let $\epsilon, b \geqslant 0$, $\lambda > 0$, $w \in \mathbb{N}$, $p = (0, 1/2)$, and $W \subset [d]$ of size w. Let $\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^d$ be an approximate solution of the expected Lasso problem $\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\boldsymbol{\beta})$ with error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1 + b\epsilon$. Then*

$$\sum_{j\in W}\left(\widetilde{\beta}_j - \frac{2p}{1+\lambda+4p^2(w-1)}\right)^2 + \sum_{j\notin W}\widetilde{\beta}_j^2 \leqslant \frac{\epsilon}{(1-\epsilon)(1-4p^2)}\left(1 + 2b + \frac{4pw(\lambda-p)}{1+\lambda+4p^2(w-1)}\right).$$

Proof. Let $\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^d}\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\boldsymbol{\beta})$ from Lemma 39. First notice that

$$\begin{aligned} &\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\widetilde{\boldsymbol{\beta}}) - \overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}}) \leqslant \lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1 + b\epsilon \leqslant \epsilon\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\widetilde{\boldsymbol{\beta}}) + b\epsilon \implies \\ &\qquad (1-\epsilon)\big(\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\widetilde{\boldsymbol{\beta}}) - \overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}})\big) \leqslant \epsilon\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}}) + b\epsilon. \end{aligned} \tag{19}$$

By expanding $\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\widetilde{\boldsymbol{\beta}})$ around $\hat{\boldsymbol{\beta}}$,

$$\begin{aligned} \frac{\epsilon}{1-\epsilon}(\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}}) + b) &\geqslant \overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\widetilde{\boldsymbol{\beta}}) - \overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}}) && \text{(by Equation (19))} \\ &= \frac{1}{2}(\widetilde{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}})^\top\nabla^2\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}})(\widetilde{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}) && \text{(by } \nabla\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}}) = \mathbf{0}) \\ &\geqslant \frac{1-4p^2}{2}\|\widetilde{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}\|_2^2. && \text{(by } \nabla^2\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}}) \geq (1-4p^2)\mathbf{I}_d) \end{aligned}$$

This finishes the proof after inserting the expression for $\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}})$ from Lemma 39. □

We are now ready to prove that approximate solutions to expected Lasso problems can be used to find good approximations to the hidden set W.

Theorem 41. *Let $\lambda > \sqrt{5/d}$, $\epsilon \in (1/d\lambda^2, \min\{1/\lambda^2, 1/5\})$, $w = \lfloor 1/(\lambda^2\epsilon)\rfloor$, $p = 1/(2\lfloor 1/(\lambda^2\epsilon)\rfloor)$, and $W \subset [d]$ of size w. Let $\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^d$ be an approximate solution of the expected Lasso problem $\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\boldsymbol{\beta})$ with error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1/400 + \epsilon/400$. Then the set $\widetilde{W} \triangleq \{i \in [d] : |\widetilde{\beta}_i| \geqslant \lambda\epsilon/2\}$ satisfies $|\widetilde{W}\Delta W| \leqslant w/5$.*

Proof. Let $\nu \triangleq \frac{2p}{1+\lambda+4p^2(w-1)}$. Since $\widetilde{\boldsymbol{\beta}}$ minimises $\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\boldsymbol{\beta})$ up to error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$, Lemma 40 yields

$$\sum_{j\in W}(\widetilde{\beta}_j-\nu)^2+\sum_{j\notin W}\widetilde{\beta}_j^2 \leqslant \frac{\epsilon/400}{(1-\epsilon/400)(1-4p^2)}\left(2+\frac{4pw(\lambda-p)}{1+\lambda+4p^2(w-1)}\right) \leqslant \frac{\epsilon}{60},$$

where we used that $\epsilon \leqslant 1/5$, $w \geqslant 1$, $p \leqslant 1/8$, and that $2pw \leqslant 1$. Therefore,

- at most $w/10$ many $j \in [d]\backslash W$ have $|\widetilde{\beta}_j| \geqslant \sqrt{\frac{\epsilon}{60}\frac{10}{w}} = \sqrt{\frac{1}{6}\frac{\epsilon}{\lfloor 1/\lambda^2\epsilon\rfloor}}$;
- at least $w - w/10$ many $j \in W$ have $\widetilde{\beta}_j \geqslant \nu - \sqrt{\frac{1}{6}\frac{\epsilon}{\lfloor 1/\lambda^2\epsilon\rfloor}}$.

Note that $\nu - \sqrt{\frac{1}{6}\frac{\epsilon}{\lfloor 1/\lambda^2\epsilon\rfloor}} = \frac{1/\lfloor 1/\lambda^2\epsilon\rfloor}{1+\lambda+1/\lfloor 1/\lambda^2\epsilon\rfloor - 1/\lfloor 1/\lambda^2\epsilon\rfloor^2} - \sqrt{\frac{1}{6}\frac{\epsilon}{\lfloor 1/\lambda^2\epsilon\rfloor}} \geqslant \frac{\lambda\epsilon}{2} \geqslant \sqrt{\frac{1}{6}\frac{\epsilon}{\lfloor 1/\lambda^2\epsilon\rfloor}}$ since it holds that $\lambda^2\epsilon \leqslant \frac{1}{\lfloor 1/\lambda^2\epsilon\rfloor} \leqslant \frac{\lambda^2\epsilon}{1-\lambda^2\epsilon} \leqslant \frac{3\lambda^2\epsilon}{4}$. This means that the set $\widetilde{W} \triangleq \{i \in [d] : |\widetilde{\beta}_i| \geqslant \lambda\epsilon/2\}$ omits at most $w/10$ indices $j \in W$ and includes at most $w/10$ indices $j \in [d]\backslash W$. Therefore $|\widetilde{W}\Delta W| \leqslant w/5$. □

The above result tells us that a Lasso algorithm that finds an approximate expected Lasso solution with respect to $\mathcal{D}_{p,W}$ can also find a set $\widetilde{W} \subset [d]$ such that $|\widetilde{W}\Delta W| \leqslant w/5$.

5.3 Quantum lower bound

We now finally piece all the results from the previous sections together in order to obtain query lower bounds for Lasso algorithms. The first step is to relate the expected Lasso $\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\boldsymbol{\beta})$ with the usual Lasso $\mathcal{L}^{(\lambda)}_{\mathbf{X},\mathbf{y}}(\boldsymbol{\beta})$ from Equation (5) considered throughout the paper. For such, we consider the normalised Lasso problem with inputs $(\mathbf{X},\mathbf{y}) \in \mathbb{R}^{n\times d}\times\mathbb{R}^n$ defined as

$$\hat{\boldsymbol{\beta}} \in \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \overline{\mathcal{L}}^{(\lambda)}_{\mathbf{X},\mathbf{y}}(\boldsymbol{\beta}) \quad \text{where} \quad \overline{\mathcal{L}}^{(\lambda)}_{\mathbf{X},\mathbf{y}}(\boldsymbol{\beta}) \triangleq \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\beta}\|_1 \quad \text{for } \lambda>0.$$

Fact 42 ([MRT18, Theorem 11.16])**.** *Let $\mathcal{D}$ be a distribution over $[-1,1]^d\times[-1,1]$. Given a sample set $\{(\mathbf{x}_i,y_i)\}_{i\in[n]}$ containing n i.i.d. samples from $\mathcal{D}$, let $\mathbf{X}\in[-1,1]^{n\times d}$ and $\mathbf{y}\in[-1,1]^n$ be such that the i-th row of $\mathbf{X}$ is $\mathbf{x}_i$ and the i-th entry of $\mathbf{y}$ is y_i. For all $\delta>0$ and $\boldsymbol{\beta}\in\mathbb{R}^d$ such that $\|\boldsymbol{\beta}\|_1\leqslant\Lambda$, with probability at least $1-\delta$ over the choice of $(\mathbf{X},\mathbf{y})$,*

$$\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}}(\boldsymbol{\beta})-\overline{\mathcal{L}}^{(\lambda)}_{\mathbf{X},\mathbf{y}}(\boldsymbol{\beta}) \leqslant \Lambda(1+\Lambda)\sqrt{\frac{2\ln(2d)}{n}}+(1+\Lambda)^2\sqrt{\frac{\ln(1/\delta)}{8n}}.$$

Using a reduction from $\mathsf{ASF}^{d,N}_{w,p}$ to algorithms for Lasso, we now prove our quantum lower bound.

Theorem 43. *Let $n<d$ such that $n=\Omega(\log d)$, $\lambda=\Omega(n)$, and $\epsilon\in(0,1/5)$ such that $\epsilon=\Omega(\sqrt{n^3\log d}/\lambda^2)$. There is $\mathbf{X}\in\{-1,1\}^{n\times d}$ such that every bounded-error quantum algorithm that computes an $\epsilon\lambda\|\widetilde{\boldsymbol{\beta}}\|_1$-minimiser for the Lasso $\mathcal{L}^{(\lambda)}_{\mathbf{X},1^n}(\boldsymbol{\beta})$ uses*

$$\Omega\left(\frac{n^3\sqrt{d}}{\lambda^3\epsilon^{3/2}}\right) \quad \textit{quantum queries to } \mathbf{X}.$$

Proof. Let $w=\lfloor 1/(\overline{\lambda}^2\epsilon)\rfloor$ and $p=1/(2\lfloor 1/(\overline{\lambda}^2\epsilon)\rfloor)$. Consider a valid input $\mathbf{X}'\in\{-1,1\}^{N\times d}$ for $\mathsf{ASF}^{d,N}_{w,p}$, and let $W\subset[d]$ be the set of w indices of columns of $\mathbf{X}'$ whose entries add up to $2pN$. According to Fact 37, we can obtain from $\mathbf{X}'$ a number of $n=\Omega(\log d)$ i.i.d. samples $\{(\mathbf{x}_i,1)\}_{i\in[n]}$

from the distribution $\mathcal{D}_{p,W}$. Let $\mathbf{X} \in \{-1,1\}^{n\times d}$ be the matrix whose rows are $\{\mathbf{x}_i\}_{i\in[n]}$. For a small enough constant $c \in (0,1)$, suppose we have a quantum Lasso algorithm which returns a $c\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$-minimiser $\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^d$ with high probability for the Lasso $\mathcal{L}^{(\lambda)}_{\mathbf{X},1^n}(\boldsymbol{\beta})$ with inputs $(\mathbf{X}, \mathbf{y} = 1^n)$. The vector $\widetilde{\boldsymbol{\beta}}$ is also a $c\overline{\lambda}\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$-minimiser to the normalised Lasso $\overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathbf{X},1^n}(\boldsymbol{\beta})$ where $\overline{\lambda} = \lambda/n$. Note that $\overline{\lambda} = \Omega(1)$. Consider now the expected Lasso $\overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathcal{D}_{p,W}}(\boldsymbol{\beta})$. According to Lemma 39, the solution $\hat{\boldsymbol{\beta}} \in \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathcal{D}_{p,W}}(\boldsymbol{\beta})$ is

$$\hat{\beta}_j = \begin{cases} \frac{2p}{1+\overline{\lambda}+4p^2(w-1)} & \text{if } j \in W, \\ 0 & \text{if } j \notin W. \end{cases} \tag{20}$$

For $n = \Omega(\epsilon^{-2}(1+\Lambda)^4 \log(d/\delta))$, with probability at least $1-\delta$ we have that

$$\overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathcal{D}_{p,W}}(\boldsymbol{\beta}) - \overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathbf{X},\mathbf{y}}(\boldsymbol{\beta}) \leqslant c\epsilon \quad \text{for all } \boldsymbol{\beta} \in \mathbb{R}^d \text{ such that } \|\boldsymbol{\beta}\|_1 \leqslant \Lambda. \tag{21}$$

Assume for now that Λ is large enough so that $\max\{\|\widetilde{\boldsymbol{\beta}}\|_1, \|\hat{\boldsymbol{\beta}}\|_1\} \leqslant \Lambda$. According to Equation (20), we know that $\|\hat{\boldsymbol{\beta}}\|_1 = \frac{2pw}{1+\overline{\lambda}+4p^2(w-1)} \leqslant \frac{1}{1+\overline{\lambda}}$. On the other hand,

$$\overline{\lambda}\|\widetilde{\boldsymbol{\beta}}\|_1 \leqslant \overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathbf{X},1^n}(\widetilde{\boldsymbol{\beta}}) \leqslant \overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathbf{X},1^n}(\hat{\boldsymbol{\beta}}) + c\overline{\lambda}\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1 \leqslant \overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}}) + c\epsilon + c\overline{\lambda}\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1,$$

since $\widetilde{\boldsymbol{\beta}}$ is a $c\overline{\lambda}\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$-minimiser and by using Equation (21). By also using that $\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}}) \leqslant \frac{1}{2} + \frac{2pw(\overline{\lambda}-p)}{1+\overline{\lambda}+4p^2(w-1)}$ according to Lemma 39, we obtain that

$$\|\widetilde{\boldsymbol{\beta}}\|_1 \leqslant \frac{1}{c\overline{\lambda}(1-\epsilon)}\left(\frac{1}{2} + \frac{2pw(\overline{\lambda}-p)}{1+\overline{\lambda}+4p^2(w-1)} + c\epsilon\right) \leqslant \frac{1}{c\overline{\lambda}} + \frac{5}{4c(1+\overline{\lambda})}. \qquad \text{(since } \epsilon \leqslant 1/5\text{)}$$

We can therefore set $\Lambda = 3/(c\overline{\lambda}) = O(1)$ in order to employ Equation (21) with $\widetilde{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\beta}}$. Note that $n = \Omega(\epsilon^{-2}(1+\Lambda)^2 \log(d/\delta))$ is satisfied within our range of parameters since $\frac{n^4}{\epsilon^2\lambda^4}\log d = O(n)$. Thus, by Equation (21) and a simple Chernoff bound in order to argue that $\overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathbf{X},\mathbf{y}}(\hat{\boldsymbol{\beta}}) - \overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}}) \leqslant c\epsilon$ (which only applies to $\hat{\boldsymbol{\beta}}$ and not all $\boldsymbol{\beta}$ such that $\|\boldsymbol{\beta}\|_1 \leqslant \Lambda$ as in Equation (21)),

$$\overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathcal{D}_{p,W}}(\widetilde{\boldsymbol{\beta}}) - \overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathcal{D}_{p,W}}(\hat{\boldsymbol{\beta}}) \leqslant 2c\epsilon + \overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathbf{X},1^n}(\widetilde{\boldsymbol{\beta}}) - \overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathbf{X},1^n}(\hat{\boldsymbol{\beta}}) \leqslant 2c\epsilon + c\overline{\lambda}\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1,$$

which means that $\widetilde{\boldsymbol{\beta}}$ is an approximate solution to $\overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathcal{D}_{p,W}}(\boldsymbol{\beta})$ with error $2c\epsilon + c\overline{\lambda}\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$. Now note that the condition $\epsilon \in (1/d\overline{\lambda}^2, \min\{1/\overline{\lambda}^2, 1/5\})$ is satisfied since $1/d\overline{\lambda}^2 = O(\sqrt{n^3 \log d}/\lambda^2)$ and $1/\overline{\lambda} = O(1)$ and $1/\overline{\lambda}^2 = n^2/\lambda^2 = \Omega(\sqrt{n^3 \log d}/\lambda^2)$. Therefore, according to Theorem 41, we can use $\widetilde{\boldsymbol{\beta}}$ in order to output $\widetilde{W} \subset [d]$ such that $|\widetilde{W}\Delta W| \leqslant w/5$ with high probability, thus solving the $\mathsf{ASF}^{d,N}_{w,p}$ problem. Since the query complexity of $\mathsf{ASF}^{d,N}_{w,p}$ is $\Omega(\sqrt{wd}/p)$ by Fact 34, this yields that the query complexity of the Lasso algorithm for $\mathcal{L}^{(\lambda)}_{\mathbf{X},1^n}(\boldsymbol{\beta})$ is

$$\Omega\left(\frac{\sqrt{wd}}{p}\right) = \Omega\left(\frac{1}{\overline{\lambda}^2\epsilon}\sqrt{\frac{d}{\overline{\lambda}^2\epsilon}}\right) = \Omega\left(\frac{n^3\sqrt{d}}{\lambda^3\epsilon^{3/2}}\right). \qquad \square$$

5.4 Classical lower bound

We now turn our attention to proving a classical query lower bound by reducing the $\mathsf{ESF}^{d,N}_{w,p}$ problem to an approximate Lasso one. The proof is similar to the quantum lower bound, the main difference being that the reduction from Fact 38 is used instead of Fact 37.

Theorem 44. *Let $n < d$ such that $n = \Omega(\log d)$, $\lambda = \Omega(n)$, and $\epsilon \in (0, 1/5)$ such that $\epsilon = \Omega(\sqrt{n^3 \log d}/\lambda^2)$. There is $\mathbf{X} \in \{-1, 1\}^{n \times d}$ such that every bounded-error classical algorithm that computes an $\epsilon\lambda\|\widetilde{\boldsymbol{\beta}}\|_1$-minimiser for the Lasso $\mathcal{L}^{(\lambda)}_{\mathbf{X},1^n}(\boldsymbol{\beta})$ uses*

$$\Omega\left(\frac{n^4 d}{\lambda^4 \epsilon^2 \log d}\right) \quad \textit{classical queries to } \mathbf{X}.$$

Proof. Let $w = \lfloor 1/(\overline{\lambda}^2\epsilon)\rfloor$ and $p = 1/(2\lfloor 1/(\overline{\lambda}^2\epsilon)\rfloor)$. Consider a valid input $\mathbf{X}' \in \{-1, 1\}^{N \times d}$ for $\mathsf{ESF}^{d,N}_{w,p}$. For a small enough constant $c \in (0, 1)$, suppose we have a classical Lasso algorithm which returns a $c\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$-minimiser $\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^d$ with high probability for the Lasso $\mathcal{L}^{(\lambda)}_{\mathbf{X},1^n}(\boldsymbol{\beta})$ with inputs $(\mathbf{X}, \mathbf{y} = 1^n)$ depending on $\mathbf{X}'$. The vector $\widetilde{\boldsymbol{\beta}}$ is also a $c\overline{\lambda}\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$-minimiser to the normalised Lasso $\overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathbf{X},1^n}(\boldsymbol{\beta})$ where $\overline{\lambda} = \lambda/n$. Similarly to the proof of Theorem 43, one can show that $\widetilde{\boldsymbol{\beta}}$ is, with high probability, an approximate solution to $\overline{\mathcal{L}}^{(\overline{\lambda})}_{\mathcal{D}_{p,W}}(\boldsymbol{\beta})$ with error $2c\epsilon + c\overline{\lambda}\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$ if $n = \Omega(\epsilon^{-2}(1 + \Lambda)^4 \log(d/\delta))$ where $\Lambda = 3/(c\overline{\lambda}) = O(1)$. This condition is satisfied within our range of parameters since $\frac{n^4}{\epsilon^2\lambda^4}\log d = O(n)$. On the other hand, again we note that the condition $\epsilon \in (1/d\overline{\lambda}^2, \min\{1/\overline{\lambda}^2, 1/5\})$ is satisfied since $1/d\overline{\lambda}^2 = O(\sqrt{n^3 \log d}/\lambda^2)$ and $1/\overline{\lambda} = O(1)$ and $1/\overline{\lambda}^2 = n^2/\lambda^2 = \Omega(\sqrt{n^3 \log d}/\lambda^2)$. Therefore, by Theorem 41, we can use the classical Lasso algorithm to solve $\mathsf{DSF}^{d,n}_{w,p}$. Finally, this means that, by Fact 38, $O(\log d)$ uses of the classical Lasso algorithm can solve $\mathsf{ESF}^{d,N}_{w,p}$. Since the query complexity of $\mathsf{ESF}^{d,N}_{w,p}$ is $\Omega((d - w)/p^2)$ by Fact 35, this yields that the query complexity of the classical Lasso algorithm for $\mathcal{L}^{(\lambda)}_{\mathbf{X},1^n}(\boldsymbol{\beta})$ is

$$\Omega\left(\frac{d - w}{p^2 \log d}\right) = \Omega\left(\frac{d - 1/(\overline{\lambda}^2\epsilon)}{\overline{\lambda}^4\epsilon^2 \log d}\right) = \Omega\left(\frac{n^4 d}{\epsilon^2\lambda^4 \log d}\right). \qquad \square$$

6 Bounds in noisy regime

In this section, we assume that the observation vector $\mathbf{y} \in \mathbb{R}^n$ and the design matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ are connected via the standard linear model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \mathbf{w},$$

where the true solution $\boldsymbol{\beta}^* \in \mathbb{R}^d$ (also referred to as regression vector) is sparse and $\mathbf{w} \in \mathbb{R}^n$ is a noise vector. Our aim is to determine how close the Lasso solution produced by the LARS algorithm, especially its approximate version from Algorithm 3, is to the true solution $\boldsymbol{\beta}^*$. More specifically, we shall focus on bounding the *mean-squared prediction error* $\|\mathbf{X}(\boldsymbol{\beta}^* - \widetilde{\boldsymbol{\beta}})\|_2^2/n$. Throughout this section we shall assume the following.

Assumption 45. *Given the design matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, then $\max_{i \in [d]} \|\mathbf{X}_i\|_2 \leqslant \sqrt{Cn}$ for some $C > 0$.*

The results from this section are standard in the Lasso literature [Wai19, BvdG11] for the case when the regularisation path from LARS algorithm exactly minimises the Lasso cost function. We generalise them for case when $\widetilde{\mathcal{P}}$ is an approximate regularisation path with error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$.

6.1 Slow rates

It is possible to obtain bounds on the mean-squared prediction error without barely any assumptions on the design matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and the noise vector $\mathbf{w} \in \mathbb{R}^n$, as the next result shows.

Theorem 46. *Let $\lambda > 0$ and $\epsilon \in [0, 1)$. Any approximate solution $\widetilde{\boldsymbol{\beta}}$ with error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$ of the Lasso with regularisation parameter $\lambda \geqslant \|\mathbf{X}^\top \mathbf{w}\|_\infty/(1-\epsilon)$ satisfies*

$$\|\mathbf{X}(\boldsymbol{\beta}^* - \widetilde{\boldsymbol{\beta}})\|_2^2 \leqslant 2(2-\epsilon)\lambda\|\boldsymbol{\beta}^*\|_1.$$

Proof. Notice first that, since $\widetilde{\boldsymbol{\beta}}$ is a minimiser up to additive error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$, then

$$\mathcal{L}_\lambda(\widetilde{\boldsymbol{\beta}}) - \mathcal{L}_\lambda(\boldsymbol{\beta}^*) \leqslant \lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1 \implies \frac{1}{2}\|\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}\|_2^2 + \lambda(1-\epsilon)\|\widetilde{\boldsymbol{\beta}}\|_1 \leqslant \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2 + \lambda\|\boldsymbol{\beta}^*\|_1.$$

Use the equality $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \mathbf{w}$ in the above inequality to obtain

$$\frac{1}{2}\|\mathbf{X}(\boldsymbol{\beta}^* - \widetilde{\boldsymbol{\beta}}) + \mathbf{w}\|_2^2 + \lambda(1-\epsilon)\|\widetilde{\boldsymbol{\beta}}\|_1 \leqslant \frac{1}{2}\|\mathbf{w}\|_2^2 + \lambda\|\boldsymbol{\beta}^*\|_1,$$

from which we get the following inequality

$$\begin{aligned}
&\frac{1}{2}\|\mathbf{X}(\boldsymbol{\beta}^* - \widetilde{\boldsymbol{\beta}})\|_2^2 + \lambda(1-\epsilon)\|\widetilde{\boldsymbol{\beta}}\|_1 + \frac{1}{2}\|\mathbf{w}\|_2^2 + \mathbf{w}^\top\mathbf{X}(\boldsymbol{\beta}^* - \widetilde{\boldsymbol{\beta}}) \leqslant \frac{1}{2}\|\mathbf{w}\|_2^2 + \lambda\|\boldsymbol{\beta}^*\|_1 \iff \\
&\qquad \frac{1}{2}\|\mathbf{X}(\boldsymbol{\beta}^* - \widetilde{\boldsymbol{\beta}})\|_2^2 + \lambda(1-\epsilon)\|\widetilde{\boldsymbol{\beta}}\|_1 + \mathbf{w}^\top\mathbf{X}(\boldsymbol{\beta}^* - \widetilde{\boldsymbol{\beta}}) \leqslant \lambda\|\boldsymbol{\beta}^*\|_1 \qquad (22)
\end{aligned}$$

and finally

$$\begin{aligned}
\frac{1}{2}\|\mathbf{X}(\boldsymbol{\beta}^* - \widetilde{\boldsymbol{\beta}})\|_2^2 &\leqslant \mathbf{w}^\top\mathbf{X}(\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) + \lambda\|\boldsymbol{\beta}^*\|_1 - \lambda(1-\epsilon)\|\widetilde{\boldsymbol{\beta}}\|_1 && \text{(Equation (22))} \\
&\leqslant \|\mathbf{X}^\top\mathbf{w}\|_\infty\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_1 + \lambda\|\boldsymbol{\beta}^*\|_1 - \lambda(1-\epsilon)\|\widetilde{\boldsymbol{\beta}}\|_1 && \text{(Hölder's inequality)} \\
&\leqslant \|\mathbf{X}^\top\mathbf{w}\|_\infty(\|\widetilde{\boldsymbol{\beta}}\| + \|\boldsymbol{\beta}^*\|_1) + \lambda\|\boldsymbol{\beta}^*\|_1 - \lambda(1-\epsilon)\|\widetilde{\boldsymbol{\beta}}\|_1 && \text{(Triangle inequality)} \\
&= \|\widetilde{\boldsymbol{\beta}}\|\left(\|\mathbf{X}^\top\mathbf{w}\|_\infty - \lambda(1-\epsilon)\right) + \|\boldsymbol{\beta}^*\|(\|\mathbf{X}^\top\mathbf{w}\|_\infty + \lambda) && \\
&\leqslant \|\boldsymbol{\beta}^*\|(\|\mathbf{X}^\top\mathbf{w}\|_\infty + \lambda) && (\|\mathbf{X}^\top\mathbf{w}\|_\infty \leqslant (1-\epsilon)\lambda) \\
&\leqslant (2-\epsilon)\lambda\|\boldsymbol{\beta}^*\|. && (\|\mathbf{X}^\top\mathbf{w}\|_\infty \leqslant (1-\epsilon)\lambda) \quad \square
\end{aligned}$$

We give some examples of the above result using common linear regression models.

Linear Gaussian model. In the classical linear Gaussian model, the design matrix is deterministic, meaning that $\mathbf{X} \in \mathbb{R}^{n \times d}$ is fixed, while the noise vector $\mathbf{w} \in \mathbb{R}^n$ has i.i.d. $\mathcal{N}(0, \sigma^2)$ entries. We then bound the probability that $\lambda \geqslant \|\mathbf{X}^\top\mathbf{w}\|_\infty/(1-\epsilon)$. Using that $\mathbf{X}_i^\top\mathbf{w}$ is a sub-Gaussian random variable with parameter $\sigma\|\mathbf{X}_i\|_2 \leqslant \sigma\sqrt{Cn}$ (Fact 4), we bound the complement of this probability:

$$\mathbb{P}\left[\|\mathbf{X}^\top\mathbf{w}\|_\infty > t\right] = \mathbb{P}\left[\max_{i\in[d]}|\mathbf{X}_i^\top\mathbf{w}| > t\right] \leqslant \sum_{i=1}^d \mathbb{P}[|\mathbf{X}_i^\top\mathbf{w}| > t] \leqslant 2d\exp\left(-\frac{t^2}{2C\sigma^2 n}\right),$$

where the first inequality follows from a union bound. Setting $t = \lambda(1-\epsilon) = \sqrt{2C\sigma^2 n\log(2d/\delta)}$, the above probability is at most δ. Consequently,

$$\mathbb{P}\left[\|\mathbf{X}^\top\mathbf{w}\|_\infty \leqslant \lambda(1-\epsilon)\right] \geqslant 1 - \delta.$$

Substituting this value of λ in Theorem 46, with probability at least $1-\delta$,

$$\frac{\|\mathbf{X}(\boldsymbol{\beta}^* - \widetilde{\boldsymbol{\beta}})\|_2^2}{n} \leqslant \frac{2(2-\epsilon)\lambda\|\boldsymbol{\beta}^*\|}{n} = \frac{2-\epsilon}{1-\epsilon}\sqrt{\frac{8C\sigma^2\log(2d/\delta)}{n}}\|\boldsymbol{\beta}^*\|.$$

We see that the mean square error vanishes by a square root of n factor as n becomes large. Without additional assumptions, this is known as a slow rate.

Compressed sensing. In compressed sensing, it is common for the design matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ to be chosen by the user, and a standard choice is the standard Gaussian matrix with i.i.d. $\mathcal{N}(0,1)$ entries. Assume further that the noise vector $\mathbf{w} \in \mathbb{R}^n$ is deterministic with $\|\mathbf{w}\|_\infty \leqslant \sigma$. Thus $\mathbf{X}_i^\top \mathbf{w}$ is a sub-Gaussian random variable with parameter $\|\mathbf{w}\|_2 \leqslant \sqrt{n}\|\mathbf{w}\|_\infty \leqslant \sqrt{n}\sigma$ (Fact 4). By following the exact same steps as the previous example (this time with $C = 1$), we conclude that, with probability at least $1 - \delta$,

$$\frac{\|\mathbf{X}(\boldsymbol{\beta}^* - \widetilde{\boldsymbol{\beta}})\|_2^2}{n} \leqslant \frac{2-\epsilon}{1-\epsilon}\sqrt{\frac{8\sigma^2 \log(2d/\delta)}{n}}\|\boldsymbol{\beta}^*\|.$$

So the mean square error also vanishes by a square root of n factor as n becomes large. We shall see next that, by imposing additional restrictions on the design matrix, faster rates can be obtained.

6.2 Fast Rates

To obtain faster rates, we need additional assumptions on the design matrix $\mathbf{X}$. The condition below originates actually in the compressed sensing literature. We assume the true solution $\boldsymbol{\beta}^*$ has a hard finite support $\mathrm{supp}(\boldsymbol{\beta}^*)$.

Definition 47 (Restricted Eigenvalue condition)**.** *For $S \subseteq [d]$ and $\zeta \in \mathbb{R}$, let*

$$\mathbb{C}_\zeta(S) \triangleq \{\boldsymbol{\Delta} \in \mathbb{R}^d : \|\boldsymbol{\Delta}_{S^c}\|_1 \leqslant \zeta\|\boldsymbol{\Delta}_S\|_1\}.$$

A matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ satisfies the Restricted Eigenvalue (RE) *condition over S with parameters (κ, ζ) if*

$$\frac{1}{n}\|\mathbf{X}\boldsymbol{\Delta}\|_2^2 \geqslant \kappa\|\boldsymbol{\Delta}\|_2^2 \quad \forall \boldsymbol{\Delta} \in \mathbb{C}_\zeta(S).$$

An interpretation of the RE condition is that it bounds away the minimum eigenvalues of the matrix $\mathbf{X}^\top\mathbf{X}$ in specific directions. Ideally, we would like to bound the curvature of the cost function $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$ in all directions, which would guarantee strong bounds on the mean square error. However, in the high-dimensional setting $d \gg n$, $\mathbf{X}^\top\mathbf{X}$ is a $d \times d$ matrix with rank at most n, so it is impossible to guarantee a positive curvature in all directions. Therefore, we must relax such stringent condition on the curvature and require that it holds only on a subset $\mathbb{C}_\zeta(S)$ of vectors. The RE condition yields the following stronger result.

Theorem 48. *Let $\lambda, \kappa > 0$ and $\epsilon \in [0, 1/4]$. Let $S \triangleq \mathrm{supp}(\boldsymbol{\beta}^*)$. Suppose $\mathbf{X} \in \mathbb{R}^{n\times d}$ satisfies the* RE *condition over S with parameters $(\kappa, 5)$. Any approximate solution $\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^d$ with error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$ of the Lasso with regularisation parameter $\lambda \geqslant 4\|\mathbf{X}^\top\mathbf{w}\|_\infty$ satisfies*

$$\|\mathbf{X}(\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\|_2^2 + 3\lambda\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_1 \leqslant 81\lambda^2\frac{|S|}{n\kappa} + 18\lambda\epsilon\|\boldsymbol{\beta}^*_S\|_1.$$

The proof has many variations, and we give the most direct one below. Note the above implies

$$\|\mathbf{X}(\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\|_2^2 \leqslant 81\lambda^2\frac{|S|}{n\kappa} + 18\lambda\epsilon\|\boldsymbol{\beta}^*_S\|_1 \quad \text{and} \quad \|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_1 \leqslant 27\lambda\frac{|S|}{n\kappa} + 6\epsilon\|\boldsymbol{\beta}^*_S\|_1.$$

Proof. We use the basic optimality of $\widetilde{\boldsymbol{\beta}}$,

$$\frac{1}{2}\|\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}\|_2^2 + \lambda(1-\epsilon)\|\widetilde{\boldsymbol{\beta}}\|_1 \leqslant \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2 + \lambda\|\boldsymbol{\beta}^*\|_1.$$

Define $\boldsymbol{\Delta} \triangleq \widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*$. Expanding out the terms like in Theorem 46,

$$\begin{aligned}
0 \leqslant 2\|\mathbf{X}\boldsymbol{\Delta}\|_2^2 &\leqslant 4\mathbf{w}^\top \mathbf{X}(\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) + 4\lambda\|\boldsymbol{\beta}^*\|_1 - 4\lambda(1-\epsilon)\|\widetilde{\boldsymbol{\beta}}\|_1 && \text{(Equation (22))} \\
&\leqslant 4\|\mathbf{X}^\top \mathbf{w}\|_\infty \|\boldsymbol{\Delta}\|_1 + 4\lambda\|\boldsymbol{\beta}^*\|_1 - 4\lambda(1-\epsilon)\|\widetilde{\boldsymbol{\beta}}\|_1 && \text{(Hölder's inequality)} \\
&\leqslant \lambda\|\boldsymbol{\Delta}\|_1 + 4\lambda\|\boldsymbol{\beta}^*\|_1 - 4\lambda(1-\epsilon)\|\widetilde{\boldsymbol{\beta}}\|_1 && (4\|\mathbf{X}^\top\boldsymbol{\Delta}\|_\infty \leqslant \lambda) \\
&\leqslant \lambda(\|\widetilde{\boldsymbol{\beta}}\|_1 + \|\boldsymbol{\beta}^*\|_1) + 4\lambda\|\boldsymbol{\beta}^*\|_1 - 4\lambda(1-\epsilon)\|\widetilde{\boldsymbol{\beta}}\|_1, && \text{(Triangle inequality)}
\end{aligned}$$

from which we infer that $\|\widetilde{\boldsymbol{\beta}}\|_1 \leqslant \frac{5}{3-4\epsilon}\|\boldsymbol{\beta}^*\|_1 \leqslant \frac{5}{2}\|\boldsymbol{\beta}^*\|_1$ since $\epsilon \leqslant \frac{1}{4}$. Let us now return to the inequality $2\|\mathbf{X}\boldsymbol{\Delta}\|_2^2 \leqslant \lambda\|\boldsymbol{\Delta}\|_1 + 4\lambda(\|\boldsymbol{\beta}^*\|_1 - \|\widetilde{\boldsymbol{\beta}}\|_1) + 4\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}\|_1$ and further decompose the right-hand side into the respective partitioning sets S and S^c by using that $\boldsymbol{\beta}^*_S + \boldsymbol{\Delta}_S = \widetilde{\boldsymbol{\beta}}_S$ and $\boldsymbol{\Delta}_{S^c} = \widetilde{\boldsymbol{\beta}}_{S^c}$, since $\boldsymbol{\beta}^*_{S^c} = \mathbf{0}$. We also use the inequality $\|\widetilde{\boldsymbol{\beta}}\|_1 \leqslant \frac{5}{2}\|\boldsymbol{\beta}^*\|_1 = \frac{5}{2}\|\boldsymbol{\beta}^*_S\|_1$. Therefore

$$\begin{aligned}
2\|\mathbf{X}\boldsymbol{\Delta}\|_2^2 &\leqslant \lambda\|\boldsymbol{\Delta}\|_1 + 4\lambda(\|\boldsymbol{\beta}^*\|_1 - \|\widetilde{\boldsymbol{\beta}}\|_1) + 10\lambda\epsilon\|\boldsymbol{\beta}^*_S\|_1 \\
&= \lambda(\|\boldsymbol{\Delta}_S\|_1 + \|\boldsymbol{\Delta}_{S^c}\|_1) + 4\lambda(\|\boldsymbol{\beta}^*\|_1 - \|\widetilde{\boldsymbol{\beta}}\|_1) + 10\lambda\epsilon\|\boldsymbol{\beta}^*_S\|_1 \\
&= \lambda(\|\boldsymbol{\Delta}_S\|_1 + \|\boldsymbol{\Delta}_{S^c}\|_1) + 4\lambda(\|\boldsymbol{\beta}^*_S\|_1 - \|\boldsymbol{\beta}^*_S + \boldsymbol{\Delta}_S\|_1 - \|\boldsymbol{\Delta}_{S^c}\|_1) + 10\lambda\epsilon\|\boldsymbol{\beta}^*_S\|_1 \\
&\leqslant \lambda(\|\boldsymbol{\Delta}_S\|_1 + \|\boldsymbol{\Delta}_{S^c}\|_1) + 4\lambda(\|\boldsymbol{\Delta}_S\|_1 - \|\boldsymbol{\Delta}_{S^c}\|_1) + 10\lambda\epsilon\|\boldsymbol{\beta}^*_S\|_1 && \text{(Equation (23))} \\
&= \lambda(5\|\boldsymbol{\Delta}_S\|_1 - 3\|\boldsymbol{\Delta}_{S^c}\|_1) + 10\lambda\epsilon\|\boldsymbol{\beta}^*_S\|_1,
\end{aligned}$$

where we used that

$$\|\boldsymbol{\beta}^*_S\|_1 - \|\boldsymbol{\beta}^*_S + \boldsymbol{\Delta}_S\|_1 \leqslant \|\boldsymbol{\beta}^*_S\|_1 - (\|\boldsymbol{\beta}^*_S\|_1 - \|\boldsymbol{\Delta}_S\|_1) = \|\boldsymbol{\Delta}_S\|_1. \tag{23}$$

We now proceed by considering two cases: (Case 1) $\|\boldsymbol{\Delta}_S\|_1 \geqslant \epsilon\|\boldsymbol{\beta}^*_S\|_1$ or (Case 2) $\|\boldsymbol{\Delta}_S\|_1 < \epsilon\|\boldsymbol{\beta}^*_S\|_1$. Therefore it must hold that either (Case 1)

$$2\|\mathbf{X}\boldsymbol{\Delta}\|_2^2 + 3\lambda\|\boldsymbol{\Delta}_{S^c}\|_1 \leqslant 15\lambda\|\boldsymbol{\Delta}_S\|_1$$

or (Case 2)

$$2\|\mathbf{X}\boldsymbol{\Delta}\|_2^2 + 3\lambda\|\boldsymbol{\Delta}_{S^c}\|_1 \leqslant 15\lambda\epsilon\|\boldsymbol{\beta}^*_S\|_1.$$

In the first case, we find that $\|\boldsymbol{\Delta}_{S^c}\|_1 \leqslant 5\|\boldsymbol{\Delta}_S\|_1 \implies \boldsymbol{\Delta} \in \mathbb{C}_5(S)$, which means that we can now apply the RE condition. More specifically, from $2\|\mathbf{X}\boldsymbol{\Delta}\|_2^2 + 3\lambda\|\boldsymbol{\Delta}_{S^c}\|_1 \leqslant 15\lambda\|\boldsymbol{\Delta}_S\|_1$ we get

$$2\|\mathbf{X}\boldsymbol{\Delta}\|_2^2 + 3\lambda\|\boldsymbol{\Delta}\|_1 \leqslant 18\lambda\|\boldsymbol{\Delta}_S\|_1 \leqslant 18\lambda\sqrt{|S|}\|\boldsymbol{\Delta}\|_2 \overset{\text{RE}}{\leqslant} 18\lambda\sqrt{\frac{|S|}{n\kappa}}\|\mathbf{X}\boldsymbol{\Delta}\|_2 \leqslant 81\lambda^2\frac{|S|}{n\kappa} + \|\mathbf{X}\boldsymbol{\Delta}\|_2^2,$$

where the last inequality follows from $2ab \leqslant a^2 + b^2$ for $a, b \in \mathbb{R}$. Thus $\|\mathbf{X}\boldsymbol{\Delta}\|_2^2 + 3\lambda\|\boldsymbol{\Delta}\|_1 \leqslant 81\lambda^2\frac{|S|}{n\kappa}$.

In the second case, from $2\|\mathbf{X}\boldsymbol{\Delta}\|_2^2 + 3\lambda\|\boldsymbol{\Delta}_{S^c}\|_1 \leqslant 15\lambda\epsilon\|\boldsymbol{\beta}^*_S\|_1$ we get

$$2\|\mathbf{X}\boldsymbol{\Delta}\|_2^2 + 3\lambda\|\boldsymbol{\Delta}\|_1 \leqslant 18\lambda\epsilon\|\boldsymbol{\beta}^*_S\|_1 \implies \|\mathbf{X}\boldsymbol{\Delta}\|_2^2 + 3\lambda\|\boldsymbol{\Delta}\|_1 \leqslant 18\lambda\epsilon\|\boldsymbol{\beta}^*_S\|_1.$$

Putting both Cases 1 and 2 together leads to the desired result. □

Linear Gaussian model and compressed sensing. Once again, we can apply the above result to common linear regression models. In the case when $\mathbf{X} \in \mathbb{R}^{n\times d}$ is fixed and $\mathbf{w} \in \mathbb{R}^n$ has i.i.d. $\mathcal{N}(0, \sigma^2)$ entries, the same reasoning from the previous section yields

$$\mathbb{P}\left[\lambda \geqslant 4\|\mathbf{X}^\top\mathbf{w}\|_\infty\right] \geqslant 1 - \delta$$

by setting $\lambda = \sqrt{32C\sigma^2 n \log(2d/\delta)}$. This value of λ leads to, with probability at least $1-\delta$,

$$\frac{\|\mathbf{X}(\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\|_2^2}{n} \leqslant \frac{2592C\sigma^2|S|\log(2d/\delta)}{\kappa n} + 18\epsilon\sqrt{\frac{32C\sigma^2\log(2d/\delta)}{\kappa n}}\|\boldsymbol{\beta}^*_S\|_1.$$

The same bound (with $C = 1$) holds for the model where $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the standard Gaussian matrix with i.i.d. $\mathcal{N}(0,1)$ entries and $\mathbf{w} \in \mathbb{R}^n$ is fixed with $\|\mathbf{w}\|_\infty \leqslant \sigma$. Thus, under the RE condition, if $\epsilon = O(\sqrt{\log(d)/n})$, then the mean square error vanishes by an n factor as n becomes large, which can be substantially smaller than the $\sqrt{\log(d)/n}$ bound from the previous section. This is known as a fast rate.

7 Discussion and future work

We studied and quantised the LARS pathwise algorithm (or homotopy method) proposed by Efron *et al.* [EHJT04] and Osborne *et al.* [OPT00a, OPT00b] which produces the set of Lasso solutions for varying the penalty term λ. By assuming quantum access to the Lasso input, we proposed two quantum algorithms. The first one (Algorithm 2) simply replaces the classical search within the joining time calculation with the quantum minimum-finding subroutine from Dürr and Høyer [DH96]. Similar to the classical LARS algorithm, it outputs the exact Lasso path, but has an improved runtime by a quadratic factor in the number of features d. Our second quantum algorithm (Algorithm 3), on the other hand, outputs an approximate Lasso path $\widetilde{\mathcal{P}}$ by computing the joining times up to some error. This is done by approximating the joining times using quantum amplitude estimation [BHMT02] and finding their maximum via the approximate quantum minimum-finding subroutine from Chen and de Wolf [CdW23]. Consequently, the runtime of our approximate quantum LARS algorithm is quadratically improved in both the number of features d and observations n. We established the correctness of Algorithm 3 by employing an approximate version of the KKT conditions and a duality gap, and showed that $\widetilde{\mathcal{P}}$ is a minimiser of the Lasso cost function up to additive error $\lambda\epsilon\|\widetilde{\boldsymbol{\beta}}(\lambda)\|_1$. Finally, we dequantised Algorithm 3 and proposed an approximate classical LARS algorithm (Algorithm 4) based on sampling.

The time complexity of our approximate LARS algorithms, both classical and quantum, directly depends on the design matrix $\mathbf{X}$ via the quantity $\|\mathbf{X}\|_{\max}\|\mathbf{X}^+_{\mathcal{A}}\|_2$, the mutual incoherence of $\mathbf{X}$, and the mutual overlap between $\mathbf{X}$ and the vector of observations $\mathbf{y}$. When $\mathbf{X}$ is a random matrix from the $\boldsymbol{\Sigma}$-Gaussian ensemble, where $\boldsymbol{\Sigma}$ is a covariance matrix with "good" properties, and $\mathbf{y}$ is sparse, we showed that these three quantities are well bounded and that Algorithm 3 and Algorithm 4 have complexities $\widetilde{O}(\sqrt{d}/\epsilon)$ and $\widetilde{O}(d/\epsilon^2)$ per iteration, respectively, which exponentially improves the dependence on n compared to the $O(nd)$ complexity of the standard LARS algorithm. The condition that $\mathbf{y}$ is sparse follows naturally from the fact that there could be data situations in which $\mathbf{X}$ being Gaussian leads to very small, in absolute magnitude, responses in the dependent variable $\mathbf{y}$. Another reason for $\mathbf{y}$ being sparse is inaccurate or incomplete measurements [CRT06]. Usually, sparsity can be artificially generated by the corresponding matrix $\mathbf{X}$ through setting some of the regression coefficients in $\boldsymbol{\beta}^*$ to zero or setting some of the components in $\mathbf{y}$ to zero. In both cases, our algorithms would be robust enough and exhibit speed up to recover the approximate original coefficient vector. Even though we only proved such result for random Gaussian matrices, we conjecture that it should also apply to other matrices.

We point out some future directions of research following our work. In Algorithms 3 and 4, the crossing times are computed exactly and the joining times are estimated up to some error. A natural extension would be to allow errors also in computing the crossing times. This would require new techniques in order to retain the path solution continuity. Another direction is to reduce

the number of iterations of the LARS algorithm. In the classical setting, Mairal and Yu [MY12] proposed an approximate LARS algorithm with a maximum number of iterations by employing first-order optimisation methods when two kinks are too close to each other. Moreover, it would be interesting to design a fully quantum LARS algorithm by using efficient quantum subroutines for matrix multiplication and matrix inversion, e.g., based on block-encoding techniques [LC19, GSLW19, CGJ19]. Also, we believe that our quantum query lower bound is not tight and can be improved, specially the dependence on ϵ (also not tight in the work of Chen and de Wolf [CdW23]).

Our algorithms are designed to work only in the fault tolerant regime. Another future direction is to study the LARS algorithm in the NISQ (Noisy Intermediate-Scale Quantum) setting [Pre18], where algorithms operate on a relatively small number of qubits and have shallow circuit depths. In addition, while we have studied the plain-vanilla LARS algorithm, other Lasso related algorithms still remain unexplored in the quantum setting. For example, fused Lasso [TSR+04, XKW+16, Hoe10], which penalises the ℓ_1-norm of both the coefficients and their successive differences, is a generalisation of Lasso with applications to support vector classifier [Gun98, RML14, SV99]. On the other hand, grouped Lasso is a generalised model for linear regression with ℓ_1 and ℓ_2-penalties [SFHT13, FHT10b]. This model is used in settings where the design matrix can be decomposed into groups of submatrices. Simon *et al.* [SFHT13] proposed a sparse grouped Lasso algorithm which finds applications in sparse graphical modelling. We believe that similar techniques employed in this work could be applied to these alternative Lasso settings.

8 Acknowledgements

JFD thanks Hanzhong Liu and Bin Yu for very useful clarifications regarding Ref. [MY12] and Lasso in general, Yanlin Chen and Ronald de Wolf for helpful clarifications regarding Ref. [CdW23], András Gilyén for Ref. [vAGGdW17], Yihui Quek for Ref. [QCR20], and Rahul Jain, Josep Lumbreras, and Marco Tomamichel for interesting discussions. JFD also thanks Iosif Pinelis for answering a question on MathOverflow [Mat24]. JFD acknowledges funding from ERC grant No. 810115-DYNASNET. DL acknowledges funding from the Latvian Quantum Initiative under EU Recovery and Resilience Facility under project No. 2.3.1.1.i.0/1/22/I/CFLA/001 in the final part of the project. CSP gratefully acknowledges Ministry of Education (MOE), AcRF Tier 2 grant (Reference No: MOE-T2EP20220-0013) for the funding of this research. This research is supported by the National Research Foundation, Singapore and A*STAR under its CQT Bridging Grant and its Quantum Engineering Programme under grant NRF2021-QEP2-02-P05.

References

[ABD+24] Jonathan Allcock, Jinge Bao, Joao F. Doriguello, Alessandro Luongo, and Miklos Santha. Constant-depth circuits for Boolean functions and quantum memory devices using multi-qubit gates. *Quantum*, 8:1530, November 2024. doi:10.22331/q-2024-11-20-1530. 12

[ADL+20] Muhammad Asim, Max Daniels, Oscar Leong, Ali Ahmed, and Paul Hand. Invertible generative models for inverse problems: mitigating representation error and dataset bias. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 399–409. PMLR, 13–18 Jul 2020. URL: https://proceedings.mlr.press/v119/asim20a.html. 3

[AGZ09] Greg W. Anderson, Alice Guionnet, and Ofer Zeitouni. *An Introduction to Random Matrices*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2009. doi:10.1017/CBO9780511801334. 11

[vAGGdW17] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Quantum SDP-solvers: Better upper and lower bounds. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 403–414, 2017. doi:10.1109/FOCS.2017.44. 13, 45

[AT16] Taylor B. Arnold and Ryan J. Tibshirani. Efficient implementations of the generalized lasso dual path algorithm. *Journal of Computational and Graphical Statistics*, 25(1):1–27, 2016. doi:10.1080/10618600.2015.1008638. 9

[BACS07] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders. Efficient quantum algorithms for simulating sparse Hamiltonians. *Communications in Mathematical Physics*, 270(2):359–371, Mar 2007. doi:10.1007/s00220-006-0150-x. 8

[BCC+14] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse Hamiltonians. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, page 283–292, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2591796.2591854. 8

[Bel24] Armando Bellante. *Quantum algorithms for sparse recovery and machine learning*. PhD thesis, Politecnico di Milano, 2024. URL: https://hdl.handle.net/10589/228972. 10

[BEM13] Mohsen Bayati, Murat A. Erdogdu, and Andrea Montanari. Estimating lasso risk and noise level. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, page 944–952, Red Hook, NY, USA, 2013. Curran Associates Inc. URL: https://dl.acm.org/doi/abs/10.5555/2999611.2999717. 2

[BvdG11] Peter Bühlmann and Sara van de Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011. doi:10.1007/978-3-642-20192-9. 2, 7, 40

[BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002. doi:10.1090/conm/305/05215. 5, 13, 44

[BJPD17] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G. Dimakis. Compressed sensing using generative models. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 537–546. PMLR, 06–11 Aug 2017. URL: https://proceedings.mlr.press/v70/bora17a.html. 3

[BL06] Jonathan Borwein and Adrian Lewis. *Convex Analysis*. Springer, 2006. doi:10.1007/978-0-387-31256-9. 16, 22

[BT19] Giorgos Borboudakis and Ioannis Tsamardinos. Forward-backward selection with early dropping. *Journal of Machine Learning Research*, 20(8):1–39, 2019. URL: http://jmlr.org/papers/v20/17-334.html. 9

[BZ22] Armando Bellante and Stefano Zanero. Quantum matching pursuit: A quantum algorithm for sparse representations. *Phys. Rev. A*, 105:022414, Feb 2022. doi:10.1103/PhysRevA.105.022414. 10

[CDS01] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001. doi:10.1137/S003614450037906X. 3

[CGJ19] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. The power of block-encoded matrix powers: Improved regression techniques via faster Hamiltonian simulation. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2019.33. 4, 9, 45

[CMP23] Shantanav Chakraborty, Aditya Morolia, and Anurudh Peduri. Quantum Regularized Least Squares. *Quantum*, 7:988, April 2023. doi:10.22331/q-2023-04-27-988. 10

[CP09] Emmanuel J. Candès and Yaniv Plan. Near-ideal model selection by ℓ_1 minimization. *The Annals of Statistics*, 37(5A):2145 – 2177, 2009. doi:10.1214/08-AOS653. 17

[CPW17] Mei Choi Chiu, Chi Seng Pun, and Hoi Ying Wong. Big data challenges of high-dimensional continuous-time mean-variance portfolio selection and a remedy. *Risk Analysis*, 37(8):1532–1549, 2017. doi:10.1111/risa.12801. 3

[CRT06] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006. doi:10.1002/cpa.20124. 3, 44

[CS20] Alex Coad and Stjepan Srhoj. Catching Gazelles with a Lasso: Big data techniques for the prediction of high-growth firms. *Small Business Economics*, 55(3):541–565, Oct 2020. doi:10.1007/s11187-019-00203-3. 3

[CWB08] Emmanuel J. Candès, Michael B. Wakin, and Stephen P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, Dec 2008. doi:10.1007/s00041-008-9045-x. 3

[CdW23] Yanlin Chen and Ronald de Wolf. Quantum Algorithms and Lower Bounds for Linear Regression with Norm Constraints. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 38:1–38:21, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2023.38. 4, 5, 8, 9, 10, 13, 23, 27, 35, 36, 44, 45

[CYGL23] Menghan Chen, Chaohua Yu, Gongde Guo, and Song Lin. Faster quantum ridge regression algorithm for prediction. *International Journal of Machine Learning and Cybernetics*, 14(1):117–124, Jan 2023. doi:10.1007/s13042-022-01526-6. 10

[DH96] Christoph Dürr and Peter Høyer. A quantum algorithm for finding the minimum. *arXiv preprint quant-ph/9607014*, 1996. doi:10.48550/arXiv.quant-ph/9607014. 5, 13, 21, 44

[DH06] D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Trans. Inf. Theor.*, 47(7):2845–2862, sep 2006. doi:10.1109/18.959265. 7

[DK08] Abhimanyu Das and David Kempe. Algorithms for subset selection in linear regression. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, page 45–54, New York, NY, USA, 2008. Association for Computing Machinery. doi:10.1145/1374376.1374384. 3

[Don06] David L. Donoho. For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59(6):797–829, 2006. doi:10.1002/cpa.20132. 17

[Dor14] A.V. Dorugade. New ridge parameters for ridge regression. *Journal of the Association of Arab Universities for Basic and Applied Sciences*, 15:94–99, 2014. doi:10.1016/j.jaubas.2013.03.005. 9

[Dos12] Charles Dossal. A necessary and sufficient condition for exact sparse recovery by ℓ_1 minimization. *Comptes Rendus Mathematique*, 350(1):117–120, 2012. doi:10.1016/j.crma.2011.12.014. 17

[EB02] M. Elad and A.M. Bruckstein. A generalized uncertainty principle and sparse representation in pairs of bases. *IEEE Transactions on Information Theory*, 48(9):2558–2567, 2002. doi:10.1109/TIT.2002.801410. 7

[EHJT04] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407 – 499, 2004. doi:10.1214/009053604000000067. 3, 4, 9, 17, 18, 44

[FHT10a] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *J Stat Softw*, 33(1):1–22, 2010. doi:10.18637/jss.v033.i01. 3

[FHT10b] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*, 2010. doi:10.48550/arXiv.1001.0736. 45

[FL10] Jianqing Fan and Jinchi Lv. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1):101, 2010. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3092303/. 3

[FNW07] Mário A. T. Figueiredo, Robert D. Nowak, and Stephen J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):586–597, 2007. doi:10.1109/JSTSP.2007.910281. 3

[FR13] Simon Foucart and Holger Rauhut. *An Invitation to Compressive Sensing*, pages 1–39. Springer New York, New York, NY, 2013. doi:10.1007/978-0-8176-4948-7_1. 18

[Fuc04] J.-J. Fuchs. Recovery of exact sparse representations in the presence of noise. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages ii–533, 2004. doi:10.1109/ICASSP.2004.1326312. 7

[Fuc05] J.J. Fuchs. Recovery of exact sparse representations in the presence of bounded noise. *IEEE Transactions on Information Theory*, 51(10):3601–3608, 2005. doi:10.1109/TIT.2005.855614. 17

[GG08] Pierre J. Garrigues and Laurent El Ghaoui. An homotopy algorithm for the lasso with online observations. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, page 489–496, Red Hook, NY, USA, 2008. Curran Associates Inc. URL: https://dl.acm.org/doi/abs/10.5555/2981780.2981841. 2

[GKZ18] Brian R. Gaines, Juhyun Kim, and Hua Zhou. Algorithms for fitting the constrained lasso. *Journal of Computational and Graphical Statistics*, 27(4):861–871, 2018. PMID: 30618485. doi:10.1080/10618600.2018.1473777. 9

[GLM08a] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Architectures for a quantum random access memory. *Phys. Rev. A*, 78:052310, Nov 2008. doi:10.1103/PhysRevA.78.052310. 5, 12

[GLM08b] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100:160501, Apr 2008. doi:10.1103/PhysRevLett.100.160501. 5, 12

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory*

of Computing, STOC 2019, page 193–204, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3313276.3316366. 8, 10, 45

[Gun98] Steve R. Gunn. Support vector machines for classification and regression. Technical Report 1, University of Southampton, 1998. URL: https://svms.org/tutorials/Gunn1998.pdf. 45

[HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009. doi:10.1103/PhysRevLett.103.150502. 8, 10

[HK70] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi:10.1080/00401706.1970.10488634. 9, 10

[HKB75] Arthur E. Hoerl, Robert W. Kannard, and Kent F. Baldwin. Ridge regression: some simulations. *Communications in Statistics*, 4(2):105–123, 1975. doi:10.1080/03610927508827232. 9

[HMZ08] Jian Huang, Shuangge Ma, and Cun-Hui Zhang. Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, 18(4):1603–1618, 2008. URL: http://www.jstor.org/stable/24308572. 7

[Hoe10] Holger Hoefling. A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006, 2010. doi:10.1198/jcgs.2010.09208. 45

[HTW15] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman and Hall/CRC, May 2015. doi:10.1201/b18401. 3

[JR23] Samuel Jaques and Arthur G. Rattew. QRAM: A survey and critique. *arXiv preprint arXiv:2305.10310*, 2023. doi:10.48550/arXiv.2305.10310. 13

[JT09] Iain M. Johnstone and D. Michael Titterington. Statistical challenges of high-dimensional data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906):4237–4253, 2009. doi:10.1098/rsta.2009.0159. 2

[Kib03] B. M. Golam Kibria. Performance of some new ridge regression estimators. *Communications in Statistics - Simulation and Computation*, 32(2):419–435, 2003. doi:10.1081/SAC-120017499. 9

[KKK08] Jinseog Kim, Yuwon Kim, and Yongdai Kim. A gradient-based optimization algorithm for LASSO. *Journal of Computational and Graphical Statistics*, 17(4):994–1009, 2008. doi:10.1198/106186008X386210. 9

[KKMR22] Jonathan Kelner, Frederic Koehler, Raghu Meka, and Dhruv Rohatgi. Lower bounds on randomly preconditioned lasso via robust sparse designs. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24419–24431. Curran Associates, Inc., 2022. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/9a8d52eb05eb7b13f54b3d9eada667b7-Paper-Conference.pdf. 2

[KM14] Vipin Kumar and Sonajharia Minz. Feature selection: a literature review. *SmartCR*, 4(3):211–229, 2014. doi:10.6029/smartcr.2014.03.007. 9

[KMTY21] Kazuya Kaneko, Koichi Miyamoto, Naoyuki Takeda, and Kazuyoshi Yoshino. Linear regression by quantum amplitude estimation and its extension to convex optimization. *Phys. Rev. A*, 104:022430, Aug 2021. doi:10.1103/PhysRevA.104.022430. 4, 9

[KP17] Iordanis Kerenidis and Anupam Prakash. Quantum Recommendation Systems. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:21, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2017.49. 10, 12, 13

[KP20] Iordanis Kerenidis and Anupam Prakash. Quantum gradient descent for linear systems and least squares. *Phys. Rev. A*, 101:022316, Feb 2020. doi:10.1103/PhysRevA.101.022316. 10

[LC19] Guang Hao Low and Isaac L. Chuang. Hamiltonian Simulation by Qubitization. *Quantum*, 3:163, July 2019. doi:10.22331/q-2019-07-12-163. 10, 45

[LDR23] Debbie Lim, João F Doriguello, and Patrick Rebentrost. Quantum algorithm for robust optimization via stochastic-gradient online learning. *arXiv preprint arXiv:2304.02262*, 2023. doi:10.48550/arXiv.2304.02262. 8

[Led01] Michel Ledoux. *The Concentration of Measure Phenomenon*. Mathematical surveys and monographs. American Mathematical Society, 2001. doi:10.1090/surv/089. 11

[LGZ16] Seth Lloyd, Silvano Garnerone, and Paolo Zanardi. Quantum algorithms for topological and geometric analysis of data. *Nature Communications*, 7(1):10138, Jan 2016. doi:10.1038/ncomms10138. 8

[LMR14] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, Sep 2014. doi:10.1038/nphys3029. 10

[LTTT14] Richard Lockhart, Jonathan Taylor, Ryan J. Tibshirani, and Robert Tibshirani. A significance test for the lasso. *The Annals of Statistics*, 42(2):413, 2014. doi:10.1214/13-AOS1175. 3

[Mao02] K.Z. Mao. Fast orthogonal forward selection algorithm for feature subset selection. *IEEE Transactions on Neural Networks*, 13(5):1218–1224, 2002. doi:10.1109/TNN.2002.1031954. 9

[Mao04] K.Z. Mao. Orthogonal forward selection and backward elimination algorithms for feature subset selection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):629–634, 2004. doi:10.1109/TSMCB.2002.804363. 9

[Mat24] MathOverflow. Orthogonal projection XX^+ from random Gaussian matrix X, 2024. Accessed on 9th June 2024. URL: https://mathoverflow.net/questions/472759/. 45

[MB06] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436 – 1462, 2006. doi:10.1214/009053606000000281. 7

[McD09] Gary C. McDonald. Ridge regression. *WIREs Computational Statistics*, 1(1):93–100, 2009. doi:10.1002/wics.14. 9

[Mec19] Elizabeth S. Meckes. *The Random Matrix Theory of the Classical Compact Groups*. Cambridge Tracts in Mathematics. Cambridge University Press, 2019. doi:10.1017/9781108303453. 11

[Mey73] Carl D. Meyer, Jr. Generalized inversion of modified matrices. *SIAM Journal on Applied Mathematics*, 24(3):315–323, 1973. doi:10.1137/0124033. 20

[MOK23] Xiangming Meng, Tomoyuki Obuchi, and Yoshiyuki Kabashima. On model selection consistency of lasso for high-dimensional ising models. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of*

Machine Learning Research, pages 6783–6805. PMLR, 25–27 Apr 2023. URL: https://proceedings.mlr.press/v206/meng23a.html. 3

[MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, second edition, 2018. URL: https://mitpress.mit.edu/9780262039406/foundations-of-machine-learning/. 38

[MS75] Donald W. Marquardt and Ronald D. Snee. Ridge regression in practice. *The American Statistician*, 29(1):3–20, 1975. doi:10.1080/00031305.1975.10479105. 9

[MS86] Vitali D. Milman and Gideon Schechtman. *Asymptotic theory of finite dimensional normed spaces: Isoperimetric inequalities in Riemannian manifolds*, volume 1200. Springer Science & Business Media, 1986. doi:10.1007/978-3-540-38822-7. 11

[MSK+19] W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019. doi:10.1073/pnas.1900654116. 3

[MY12] Julien Mairal and Bin Yu. Complexity analysis of the lasso regularization path. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 1835–1842, Madison, WI, USA, 2012. Omnipress. URL: https://dl.acm.org/doi/10.5555/3042573.3042807. 6, 7, 9, 15, 17, 18, 22, 23, 45

[OPT00a] Michael R. Osborne, Brett Presnell, and Berwin A. Turlach. On the LASSO and its dual. *Journal of Computational and Graphical Statistics*, 9(2):319–337, 2000. doi:10.1080/10618600.2000.10474883. 4, 9, 17, 18, 44

[OPT00b] MR Osborne, B Presnell, and BA Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20(3):389–403, 07 2000. doi:10.1093/imanum/20.3.389. 4, 9, 18, 44

[PL22] Yiming Peng and Vadim Linetsky. Portfolio selection: A statistical learning approach. In *Proceedings of the Third ACM International Conference on AI in Finance*, ICAIF '22, page 257–263, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3533271.3561707. 3

[Pra14] Anupam Prakash. *Quantum algorithms for linear algebra and machine learning*. University of California, Berkeley, 2014. URL: https://escholarship.org/uc/item/5v9535q4. 8, 10, 12, 13

[Pre18] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018. doi:10.22331/q-2018-08-06-79. 45

[PW19] Chi Seng Pun and Hoi Ying Wong. A linear programming model for selection of sparse high-dimensional multiperiod portfolios. *European Journal of Operational Research*, 273(2):754–771, 2019. doi:10.1016/j.ejor.2018.08.025. 3

[QCR20] Yihui Quek, Clement Canonne, and Patrick Rebentrost. Robust quantum minimum finding with an application to hypothesis selection. *arXiv preprint arXiv:2003.11777*, 2020. doi:10.48550/arXiv.2003.11777. 13, 45

[RCC+22] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1 – 85, 2022. doi:10.1214/21-SS133. 3

[RML14] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Phys. Rev. Lett.*, 113:130503, Sep 2014. doi:10.1103/PhysRevLett.113.130503. 45

[Ros04] Saharon Rosset. Following curved regularized optimization solution paths. In *Proceedings of the 17th International Conference on Neural Information Pro-*

cessing Systems, NIPS'04, page 1153–1160, Cambridge, MA, USA, 2004. MIT Press. URL: https://proceedings.neurips.cc/paper_files/paper/2004/file/32b991e5d77ad140559ffb95522992d0-Paper.pdf. 17

[Rot04] V. Roth. The generalized LASSO. *IEEE Transactions on Neural Networks*, 15(1):16–28, 2004. doi:10.1109/TNN.2003.809398. 9

[RS14] Matthias Reif and Faisal Shafait. Efficient feature size reduction via predictive forward selection. *Pattern Recognition*, 47(4):1664–1673, 2014. doi:10.1016/j.patcog.2013.10.009. 9

[RZ07] Saharon Rosset and Ji Zhu. Piecewise linear regularized solution paths. *The Annals of Statistics*, 35(3):1012 – 1030, 2007. doi:10.1214/009053606000001370. 9, 15, 18, 20

[SCL+18] Karl Sjöstrand, Line Harder Clemmensen, Rasmus Larsen, Gudmundur Einarsson, and Bjarne Kjær Ersbøll. SpaSM: A MATLAB toolbox for sparse statistical modeling. *Journal of Statistical Software*, 84(10), 2018. doi:10.18637/jss.v084.i10. 15, 18

[SE20] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. URL: https://dl.acm.org/doi/abs/10.5555/3495724.3496767. 3

[SFHT13] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013. doi:10.1080/10618600.2012.681250. 45

[SGV98] Craig Saunders, Alexander Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 515–521, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. URL: https://dl.acm.org/doi/10.5555/645527.657464. 9

[Sha23] Changpeng Shao. Quantum speedup of leverage score sampling and its application. *arXiv preprint arXiv:2301.06107*, 2023. doi:10.48550/arXiv.2301.06107. 4, 10

[Sho94] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994. doi:10.1109/SFCS.1994.365700. 4

[Sho99] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999. doi:10.1137/S0036144598347011. 4

[SSP16] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. Prediction by linear regression on a quantum computer. *Phys. Rev. A*, 94:022342, Aug 2016. doi:10.1103/PhysRevA.94.022342. 10

[Sto13] Mihailo Stojnic. A framework to characterize performance of LASSO algorithms. *arXiv preprint arXiv:1303.7291*, 2013. doi:10.48550/arXiv.1303.7291. 9

[SV99] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, Jun 1999. doi:10.1023/A:1018628609742. 45

[SX20] Changpeng Shao and Hua Xiang. Quantum regularized least squares solver with parameter estimate. *Quantum Information Processing*, 19(4):113, Feb 2020. doi:10.1007/s11128-020-2615-9. 4, 10

[TFZB08] Feng Tan, Xuezheng Fu, Yanqing Zhang, and Anu G. Bourgeois. A genetic algorithm-based method for feature subset selection. *Soft Computing*, 12(2):111–120, Jan 2008. doi:10.1007/s00500-007-0193-8. 9

[Tib13] Ryan J. Tibshirani. The lasso problem and uniqueness. *Electronic Journal of Statistics*, 7:1456 – 1490, 2013. doi:10.1214/13-EJS815. 5, 9, 15, 16, 17, 18

[Tib18] Robert Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 12 2018. doi:10.1111/j.2517-6161.1996.tb02080.x. 3

[Tro06] J.A. Tropp. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006. doi:10.1109/TIT.2005.864420. 7

[TSR+04] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and Smoothness Via the Fused Lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(1):91–108, 12 2004. doi:10.1111/j.1467-9868.2005.00490.x. 45

[TYG15] Shaonan Tian, Yan Yu, and Hui Guo. Variable selection and corporate bankruptcy forecasts. *Journal of Banking & Finance*, 52:89–100, 2015. doi:10.1016/j.jbankfin.2014.12.003. 3

[UGH09] M. Graziano Usai, Mike E. Goddard, and Ben J. Hayes. LASSO with cross-validation for genomic selection. *Genetics Research*, 91(6):427–436, 2009. doi:10.1017/S0016672309990334. 3

[Ver18] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018. doi:10.1017/9781108231596. 11, 34

[Vin78] Hrishikesh D. Vinod. A survey of ridge regression and related techniques for improvements over ordinary least squares. *The Review of Economics and Statistics*, 60(1):121–131, 1978. URL: http://www.jstor.org/stable/1924340. 9

[VK05] Dimitrios Ververidis and Constantine Kotropoulos. Sequential forward feature selection with low computational cost. In *2005 13th European Signal Processing Conference*, pages 1–4, 2005. doi:10.5281/ZENODO.39057. 9

[Wai09] Martin J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202, 2009. doi:10.1109/TIT.2009.2016018. 7, 17, 32, 33

[Wai19] Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019. doi:10.1017/9781108627771. 7, 32, 40

[Wan17] Guoming Wang. Quantum algorithm for linear regression. *Phys. Rev. A*, 96:012335, Jul 2017. doi:10.1103/PhysRevA.96.012335. 4, 9

[WB07] Hua-liang Wei and Stephen A. Billings. Feature subset selection and ranking for data dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):162–166, 2007. doi:10.1109/TPAMI.2007.250607. 9

[WBL12] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum algorithm for data fitting. *Phys. Rev. Lett.*, 109:050505, Aug 2012. doi:10.1103/PhysRevLett.109.050505. 4, 9

[WCH+09] Tong Tong Wu, Yi Fang Chen, Trevor Hastie, Eric Sobel, and Kenneth Lange. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, 01 2009. doi:10.1093/bioinformatics/btp041. 3

[WFL00] D. C. Whitley, M. G. Ford, and D. J. Livingstone. Unsupervised forward selection: a method for eliminating redundant variables. *Journal of Chemical Information and Computer Sciences*, 40(5):1160–1168, Sep 2000. doi:10.1021/ci000384c. 9

[vW15] Wessel N. van Wieringen. Lecture notes on ridge regression. *arXiv preprint arXiv:1509.09169*, 2015. doi:10.48550/arXiv.1509.09169. 9

[WM22] John Wright and Yi Ma. *High-dimensional data analysis with low-dimensional models: Principles, computation, and applications*. Cambridge University Press, 2022. doi:10.1017/9781108779302. 2

[XKW+16] Bo Xin, Yoshinobu Kawahara, Yizhou Wang, Lingjing Hu, and Wen Gao. Efficient generalized fused lasso and its applications. *ACM Trans. Intell. Syst. Technol.*, 7(4), may 2016. doi:10.1145/2847421. 45

[YGW21] Chao-Hua Yu, Fei Gao, and Qiao-Yan Wen. An improved quantum algorithm for ridge regression. *IEEE Transactions on Knowledge and Data Engineering*, 33(3):858–866, 2021. doi:10.1109/TKDE.2019.2937491. 4, 10

[ZJ96] D. Zongker and A. Jain. Algorithms for feature selection: An evaluation. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 2, pages 18–22 vol.2, 1996. doi:10.1109/ICPR.1996.546716. 9

[ZLZ18] Tuo Zhao, Han Liu, and Tong Zhang. Pathwise coordinate optimization for sparse learning: Algorithm and theory. *The Annals of Statistics*, 46(1):180 – 218, 2018. doi:10.1214/17-AOS1547. 3

[ZY06] Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7(90):2541–2563, 2006. URL: http://jmlr.org/papers/v7/zhao06a.html. 2, 3, 7

A Summary of symbols

The symbols and their corresponding concept are summarised in the table below.

Symbol	Explanation
n	Number of observations or sample points
d	Number of features
$\mathbf{y}$	Vector of observations
$\mathbf{X}$	Design matrix
$\boldsymbol{\beta}$	Optimisation variables
$\hat{\boldsymbol{\beta}}$	Optimal Lasso solution
$\widetilde{\boldsymbol{\beta}}$	Approximate Lasso solution
λ	Regularisation or penalty parameter
$\mathcal{L}^{(\lambda)}_{\mathbf{X},\mathbf{y}}(\boldsymbol{\beta})$	Lasso function with input $(\mathbf{X},\mathbf{y})$
$\overline{\mathcal{L}}^{(\lambda)}_{\mathbf{X},\mathbf{y}}(\boldsymbol{\beta})$	Normalised lasso function with input $(\mathbf{X},\mathbf{y})$
$\overline{\mathcal{L}}^{(\lambda)}_{\mathcal{D}}(\boldsymbol{\beta})$	Expected Lasso function relative to distribution $\mathcal{D}$
$\mathcal{P}$	Optimal regularisation Path
$\widetilde{\mathcal{P}}$	Approximate regularisation Path
$\mathcal{A}$	Active set
$\mathcal{I}$	Inactive set
$\boldsymbol{\eta}$	Equicorrelation signs
$\alpha_{\mathcal{A}}$	Mutual incoherence
$\gamma_{\mathcal{A}}$	Mutual overlap

Table 3: Summary of symbols and their corresponding concept used in the paper.