

---

# Encrypted accelerated least squares regression

---

**Pedro M. Esperança**  
Department of Statistics  
University of Oxford

**Louis J. M. Aslett**  
Department of Statistics  
University of Oxford

**Chris C. Holmes**  
Department of Statistics  
University of Oxford

## Abstract

Information that is stored in an encrypted format is, by definition, usually not amenable to statistical analysis or machine learning methods. In this paper we present detailed analysis of coordinate and accelerated gradient descent algorithms which are capable of fitting least squares and penalised ridge regression models, using data encrypted under a fully homomorphic encryption scheme. Gradient descent is shown to dominate in terms of encrypted computational speed, and theoretical results are proven to give parameter bounds which ensure correctness of decryption. The characteristics of encrypted computation are empirically shown to favour a non-standard acceleration technique. This demonstrates the possibility of approximating conventional statistical regression methods using encrypted data without compromising privacy.

## 1 Introduction

Issues surrounding data security and privacy of personal information are of growing concern to the public, governments and commercial sectors. Privacy concerns can erode confidence in the ability of organisations to store data securely, with a consequence that individuals may be reticent to contribute their personal information to scientific studies or to commercial organisations (Sanderson et al., 2015; Naveed et al., 2015; Gymrek et al., 2013).

In this paper we demonstrate that statistical regression methods can be applied directly to encrypted data without compromising privacy. Our work in-

volves adapting existing methodology for regression analysis in such a way as to enable full computation within the mathematical and computational constraints of recently developed fully homomorphic encryption schemes (Gentry, 2010; Fan and Vercauteren, 2012). We show empirically that traditional state-of-the-art convergence acceleration techniques can underperform when such constraints are taken into account.

Fully homomorphic encryption (FHE) differs from differential privacy (DP) in that it provides for exact computation with cryptographically strong privacy of all the data during the fitting process itself, at the expense of much greater restrictions on the operations and computational cost (Gentry, 2010; Aslett et al., 2015a). However, it is complementary to DP (as opposed to competing with it) in the sense that FHE maintains privacy *during* model fitting and prediction, while DP can ensure privacy post-processing of the data if the model itself is to be made public. For an overview of DP see Dwork and Roth (2014).

FHE allows for secure operations to be performed on data, and statistical analysis and machine learning is the major reason why people want to perform mathematical operations on data. Thus, there is a real opportunity for machine learning scientists to be involved in shaping the research agenda in FHE (United States EOP, 2014). The applications of encrypted statistics and machine learning include general purpose cloud computing when privacy concerns exist, and are especially important in e-health and clinical decision support (Basilakis et al., 2015; McLaren et al., 2016).

§2 is a brief accessible introduction to FHE and §3 recaps regression to fix notation and our method of representing data prior to encryption. A detailed examination of coordinate and gradient descent methods in an encrypted context follows in §4, including encrypted scaling for correctness, computational considerations, prediction, inference, regularisation and theoretical proofs for parameters in a popular FHE scheme. §5 discusses acceleration methods optimal for encrypted computation, with examples provided in §6 and discussion in §7.

## 2 Homomorphic encryption

A common technique for ensuring the privacy of data is to encrypt it, but typically once one wishes to fit a model it is necessary to first decrypt and risk exposing the data. However, recent advances in cryptography enable a very limited amount of computation to be performed directly on the encrypted content, rendering the correct result upon decryption.

### 2.1 Standard cryptography

A public key encryption scheme is one which has two algorithms,  $\text{Enc}(\cdot, \cdot)$  and  $\text{Dec}(\cdot, \cdot)$ , to perform encryption and decryption respectively, together with two keys: the public key,  $k_p$ , can be widely distributed and used by anyone to encrypt a message; the secret key,  $k_s$ , is required to decrypt any message encrypted using  $k_p$  and so is kept private. The fundamental identity is:

$$\text{Dec}(k_s, \text{Enc}(k_p, m)) = m \quad \forall m.$$

The data to be encrypted,  $m$ , is referred to as the *message* or *plaintext* and after encryption is referred to as the *ciphertext*. In conventional encryption algorithms manipulation of the ciphertext does not typically lead to meaningful modification of the message.

### 2.2 Fully Homomorphic encryption

An encryption scheme is said to be *fully homomorphic* if it also possesses two operations,  $\oplus$  and  $\otimes$ , which satisfy the following identities:

$$\begin{aligned} \text{Dec}(k_s, \text{Enc}(k_p, m_1) \oplus \text{Enc}(k_p, m_2)) &= m_1 + m_2 \\ \text{Dec}(k_s, \text{Enc}(k_p, m_1) \otimes \text{Enc}(k_p, m_2)) &= m_1 \times m_2 \end{aligned}$$

for all  $m_1, m_2$ , which can be applied a theoretically arbitrary number of times. In other words, a homomorphic encryption scheme allows computation directly on ciphertexts which will correctly decrypt the result as if the corresponding operations had been applied to the original messages.

However there are many constraints in practical implementation, reviewed in Aslett et al. (2015a). For the purposes of this work they may be synthesised as:

- Typically  $m$  can only easily represent binary or integers.
- Data size grows substantially after encryption.
- The computational cost of  $\oplus$  and  $\otimes$  is orders of magnitude higher than standard  $+$  and  $\times$ .
- Operations such as comparisons ( $=, <, >$ ) and division ( $\div$ ) are not possible.
- Implementation of current schemes necessitate a highly computationally expensive ‘bootstrap’ operation (unrelated to statistical bootstrap) which

must be applied frequently between  $\otimes$  operations to control the noise in the ciphertext.

Consequently, a crucial property of any FHE scheme is the *multiplicative depth*<sup>1</sup> which is possible before a ‘bootstrap’ is required. Typically, one can select the parameters of the FHE scheme to support a pre-specified multiplicative depth, but this is a trade-off because parameters supporting greater depth between ‘bootstraps’ also result in larger ciphertexts and slower homomorphic operations ( $\oplus, \otimes$ ). Therefore, it is essential to consider the *Maximum Multiplicative Depth* (MMD) required to evaluate an algorithm encrypted, since this dramatically affects speed and memory usage. Indeed, cryptographic parameters are typically chosen to match the MMD of the target algorithm being run so as to avoid ‘bootstrap’ altogether (which is simply deemed too computationally costly).

The above constraints often mean that standard statistical methodology cannot be applied unmodified on encrypted content.

### 2.3 Privacy preserving statistics

There has been some work towards using FHE schemes for statistics and machine learning. Often this has involved identifying existing algorithms which can be run with minimal modifications (Wu and Haven, 2012; Graepel et al., 2013), or fitted on unencrypted data for later prediction on new encrypted data (Dowlin et al., 2016). However, some recent work has also begun on developing new methodology inspired by traditional techniques, specifically tailored to homomorphic computation so that the whole analysis — model fitting and prediction — can be computed encrypted (Aslett et al., 2015b).

In particular, the topic of linear regression in the context of FHE has not been covered systematically in the literature thus far. Hall et al. (2011) propose protocols for regression analysis which involve substantial communication and intermediate decryption between multiple parties: it takes two days to complete a problem of size  $N = 51,016$  observations and  $P = 23$  predictors. In this work, we want to develop methods capable of fitting and prediction without any intermediate communication or decryption. Wu and Haven (2012) were the first to tackle linear regression in the context of FHE by using Cramer’s rule for matrix inversion. Unfortunately, this approach calls for the computation of the determinant of  $\mathbf{X}^T \mathbf{X}$ , which

<sup>1</sup>Having only addition and multiplication operations mean all computations form polynomials. Simply put, the multiplicative depth corresponds to the degree of the maximal degree term of the polynomial minus one. e.g.,  $\sum_{j=1}^P a_j b_j$  has depth 1 and  $\prod_{j=1}^P a_j$  has depth  $P - 1$ .

quickly becomes intractable for even low dimensional problems (e.g.,  $P > 4$ ). Specifically, the multiplicative depth is unbounded with growing dimension and, consequently, bootstrapping seems unavoidable. Graepel et al. (2013) mention that regression can be done by gradient descent, but do not implement the method or give further details.

The approach we present in this work enjoys several notable properties: (i) estimation and prediction can both be performed in the encrypted domain; (ii) bootstrapping can be avoided even for moderately large problems, (iii) scales linearly with the number of predictors; (iv) and admits the option of  $L_2$  (ridge) regularisation.

### 3 The linear regression model

The standard linear regression model assumes

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_N, \sigma^2 \mathbf{I}_{NN}) \quad (1)$$

where  $\mathbf{y}$  is a response vector of length  $N$ ;  $\mathbf{X}$  is a design matrix of size  $N \times P$ ;  $\boldsymbol{\beta}$  is a parameter vector of length  $P$ ; and  $\boldsymbol{\varepsilon}$  is a vector of length  $N$  of independent and Normally distributed errors with zero mean and constant variance  $\sigma^2$ . Provided that  $\mathbf{X}^T \mathbf{X}$  is invertible, the ordinary least squares (OLS) solution to

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad (2)$$

$$\text{is } \hat{\boldsymbol{\beta}}_{\text{ols}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (3)$$

Regularisation techniques trade the unbiasedness of OLS for smaller parameter variance by adding a constraint of the form  $\sum_{j=1}^P |\beta_j|^\gamma \leq c$ , for some  $\gamma$  (Hastie et al., 2009). Bounding the norm of the parameter vector imposes a penalty for complexity, resulting in shrinkage of the regression coefficients towards zero. We focus on  $L_2$  regularisation (ridge, henceforth RLS; Hoerl and Kennard, 1970), where  $\gamma = 2$ . Other options are available, although no one method seems to dominate the others (Zou and Hastie, 2005; Tibshirani, 1996; Fu, 1998). The standard solution to the regularised problem

$$\min_{\boldsymbol{\beta}} \{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \alpha \|\boldsymbol{\beta}\|_2^2 \} \quad (4)$$

$$\text{is } \hat{\boldsymbol{\beta}}_{\text{rls}}(\alpha) = (\mathbf{X}^T \mathbf{X} + \alpha \mathbf{I}_{PP})^{-1} \mathbf{X}^T \mathbf{y} \quad (5)$$

revealing that  $\alpha$  is also key in converting ill-conditioned problems into well-conditioned ones (Hoerl and Kennard, 1970).

#### 3.1 Data representation and encoding

Because FHE schemes can only naturally encrypt representations of integers, dealing with non-integer data

requires a special encoding. We use the transformation  $\dot{z} \equiv \lfloor 10^\phi z \rfloor \in \mathbb{Z}$ , for  $z \in \mathbb{R}$  and  $\phi \in \mathbb{N}$ . Here,  $\phi$  represents the desired level of accuracy or, more precisely, the number of decimal places to be retained. By construction,  $\dot{z} \approx 10^\phi z$ , so that smooth relative distances between elements encoded in this way are approximately preserved. The encoding can accommodate both discrete (integer-valued) data and an approximation to continuous (real-valued) data. Throughout, we assume that covariates are standardised and responses centred before integer encoding and encryption.

## 4 Least squares via descent methods

Conventional solutions to (3) and (5) can be found in closed form using standard matrix inversion techniques with runtime complexity  $\mathcal{O}(P^3)$  for the matrix inversion and  $\mathcal{O}(NP^2)$  for the matrix product. This direct approach was the one advocated for encrypted regression by Wu and Haven (2012), though dimensionality is constrained by school-book matrix inversion to enable homomorphic computation.

### 4.1 Iterative methods

We analyse two variants of an iterative algorithm, one which updates all parameters in  $\boldsymbol{\beta}^{[k]}$  *simultaneously* at each iteration  $k$ , using the vector  $\boldsymbol{\beta}^{[k-1]}$  (ELS-GD); and another which updates these parameters *sequentially*, using always the most current estimate for each parameter (ELS-CD).

The sequential update mode is related to the Gauss-Seidel method (coordinate descent), while the simultaneous update mode is related to the Jacobi method (gradient descent). See Varga (2000, chapter 3); Björck (1996, chapter 7).

As we will see below, there are two competing concerns here: optimisation efficiency and computational tractability within the cryptographic constraints, with these two types of updates having different properties. We demonstrate that the properties of these methods in the encrypted domain means some standard optimality results no longer apply.

#### 4.1.1 Sequential updates via coordinate descent

Standard coordinate descent (see e.g. Wright, 2015) for linear regression has the update:

$$\beta_{j_k}^{[k]} = \beta_{j_k}^{[k-1]} + \frac{\mathbf{X}_{\cdot j_k}^T (\mathbf{y} - \mathbf{X} \boldsymbol{\beta}^{\text{old}})}{\mathbf{X}_{\cdot j_k}^T \mathbf{X}_{\cdot j_k}} \quad (6)$$

where  $\boldsymbol{\beta}^{\text{old}}$  contains the components updated on the previous iteration or this one as appropriate, there

being many variants for the schedule of coordinates to update. However, this cannot be computed encrypted because the required data-dependent division is not feasible. In an alternative variant we can replace this with a generic step-size  $\delta$  and at each iteration  $k$  choose one variable, say  $j_k \in \mathbb{N}_{1:P}$ , and update the corresponding regression parameter:

$$\beta_{j_k}^{[k]} = \beta_{j_k}^{[k-1]} + \delta \mathbf{X}_{j_k}^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^{\text{old}}) \quad (7)$$

Two things are noteworthy: first, these equations require only evaluation of polynomials; and second, a rescaling is necessary since only integer polynomial functions can be computed homomorphically, and the parameter  $\delta$  is usually not an integer. We show how to perform this type of rescaling in the context of §4.1.2.

**Encrypted computation** As each update uses the most recent estimate of every parameter, the MMD grows by 2 with each parameter update (due to the term  $\mathbf{X}_{j_k}^T \mathbf{X}\boldsymbol{\beta}^{\text{old}}$ ). This implies that for  $K$  iterations over  $P$  covariates the MMD is equal to  $2KP$ . This renders the algorithm very expensive computationally, as it requires bootstrapping of ciphertexts for problems with even moderately large  $P$ .

#### 4.1.2 Simultaneous updates via gradient descent

In the case of gradient descent, for the objective function  $S(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$  with  $\nabla S(\boldsymbol{\beta}) = \partial S(\boldsymbol{\beta})/\partial \boldsymbol{\beta} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ , the update equations are:

$$\boldsymbol{\beta}^{[k]} = \boldsymbol{\beta}^{[k-1]} - \delta \nabla S(\boldsymbol{\beta}^{[k-1]}) \quad (8)$$

$$= \boldsymbol{\beta}^{[k-1]} + \delta \mathbf{X}^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^{[k-1]}). \quad (9)$$

As with coordinate descent, these updates can be computed homomorphically, although the update equations must again be rescaled. This is strictly necessary to accommodate the transformed data, in order to overcome the fact that we cannot divide (see §3.1). Letting  $\delta \equiv 1/\nu$  for  $\nu \in \mathbb{N}$ , the transformed equations for the simultaneous updates are:

$$\begin{aligned} \tilde{\boldsymbol{\beta}}^{[k]} &\equiv 10^{\phi} \tilde{\nu} \tilde{\boldsymbol{\beta}}^{[k-1]} + \tilde{\mathbf{X}}^T (10^{k\phi} \tilde{\nu}^{k-1} \tilde{\mathbf{y}} - \tilde{\mathbf{X}} \tilde{\boldsymbol{\beta}}^{[k-1]}) \\ &= 10^{(2k+1)\phi} \nu^k \boldsymbol{\beta}^{[k]} \end{aligned} \quad (10)$$

where now all transformed variables are represented with tildes, e.g.,  $\tilde{\mathbf{X}} = 10^{\phi} \mathbf{X}$  and similarly for the other variables, except the coefficients  $\{\boldsymbol{\beta}^{[k]}\}_{k \in \mathbb{N}_{0:K}}$  as their scaling is iteration dependent (see supplementary materials, §1). The rescaling factors are independent of the data and known a priori, and so can be grouped during computation, e.g.,  $10^{k\phi} \tilde{\nu}^{k-1}$  in (10) can be encrypted as a single value.

Retrieval of the coefficients can be done by the secret key holder by computing  $\boldsymbol{\beta}^{[K]} \leftarrow$

$\text{Dec}(k_s, \tilde{\boldsymbol{\beta}}^{[K]}) / (10^{(2K+1)\phi} \nu^K)$ . Note the important difference between coordinate and gradient descent: for  $K$  iterations, in CD each coefficient is updated  $K/P$  times while in GD each coefficient is updated  $K$  times.

**Encrypted computation** There is a crucial difference between CD and GD in the encrypted domain: GD reduces the multiplicative depth from  $2KP$  to  $2K$ , which is independent of  $P$ , enabling scalability to higher dimensional models without bootstrapping or having to select parameters which support greater MMD. As discussed in §2.2, bootstrapping in current FHE schemes is to be avoided wherever possible because it is very computationally expensive. This is an interesting and important result, because it means that in the specific setting of encrypted computation, Nesterov's faster rates of convergence (Nesterov, 2012) compared to gradient descent in a randomised coordinate descent setting will not apply, as we will show.

For these computational reasons we focus primarily on gradient descent hereinafter. For convergence in a regression setting recall:

**Lemma 1** (Convergence of ELS-GD). *Define  $\boldsymbol{\beta}^{[k+1]} \equiv \boldsymbol{\beta}^{[k]} + \delta \mathbf{X}^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^{[k]})$  and let  $\boldsymbol{\beta}^{[0]} \equiv \mathbf{0}_P$ . Then,  $\lim_{k \rightarrow \infty} \boldsymbol{\beta}^{[k]} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = \hat{\boldsymbol{\beta}}_{ols}$  for  $\delta \in (0, 2/S(\mathbf{X}^T \mathbf{X}))$  where  $S(\mathbf{X}^T \mathbf{X})$  is the spectral radius of  $\mathbf{X}^T \mathbf{X}$ .*

The optimal choice of step size, in the sense that it minimises the spectral radius, is  $\delta_{\star} = 2/(\lambda_{\max} + \lambda_{\min})$ , implying an optimal spectral radius  $\mathcal{S}_{\star} = (\lambda_{\max} - \lambda_{\min})/(\lambda_{\max} + \lambda_{\min})$ , where  $\lambda_{\max}$  and  $\lambda_{\min}$  denote the largest and smallest eigenvalues of  $\mathbf{X}^T \mathbf{X}$ , respectively (see Ryaben'kii and Tsynkov, 2006, Theorem 6.3 for all proofs).

**Lemma 2** (Oscillatory nature of ELS-GD). *The iterative process (8) can be written as an oscillating sum:*

$$\boldsymbol{\beta}^{[k]} = \sum_{n=1}^k (-1)^{n+1} \binom{k}{k-n} \delta^n (\mathbf{X}^T \mathbf{X})^{n-1} \mathbf{X}^T \mathbf{y} \quad (11)$$

This lemma is proved in the supplementary materials (§3). We show in §5.2 that it is possible to improve the convergence rate by using acceleration methods that exploit the oscillatory nature of the GD algorithm to accelerate the convergence of the series  $\{\boldsymbol{\beta}^{[k]}\}_{k \geq 0}$ .

## 4.2 Prediction

Note that the form of the GD equation (10) implies a common scaling factor,  $10^{(2k+1)\phi} \nu^K$ , for all parameters. Performing encrypted prediction is then straightforward as it requires only the computation of the dot product  $\tilde{y}_i = \tilde{\mathbf{X}}_i^T \tilde{\boldsymbol{\beta}}^{[K]} = 10^{(2K+1)\phi} \nu^K \hat{y}_i$ , where

$\hat{y}_i = \mathbf{X}_i^T \boldsymbol{\beta}^{[K]}$  denotes the predicted values in the space of the original data and  $\tilde{y}_i$  denotes the corresponding transformed version. Upon decryption, rescaling can be done as before by the secret key holder. The procedure increases the MMD of the algorithm by 1.

The situation is more complex for coordinate descent since at the end of the final iteration each element of  $\boldsymbol{\beta}^{[K]}$  will have different scaling. Therefore the scaling must be unified before prediction, adding additional overhead.

### 4.3 Inference

Inference in the linear regression model (e.g., confidence intervals, hypothesis testing, variable selection) requires knowledge of the standard errors of the regression coefficients:

$$\mathbb{V}[\hat{\boldsymbol{\beta}}_{\text{ols}}] = \hat{\sigma}^2 (\mathbf{X}^T \mathbf{X})^{-1}, \quad \hat{\sigma}^2 = \mathbf{e}^T \mathbf{e} / (N - P). \quad (12)$$

However, the matrix inversion is intractable under homomorphic computation except for very small  $P$ . An alternative is to estimate the standard errors by bootstrapping the data and using the variability in the parameter estimates obtained.

### 4.4 Regularisation

$L_2$ -regularised (ridge) least squares is easy to implement using the well known data augmentation procedure (Allen, 1974):

$$\hat{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \sqrt{\alpha} \mathbf{I}_{PP} \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_P \end{bmatrix}. \quad (13)$$

OLS estimates when using the augmented data,  $(\hat{\mathbf{X}}, \hat{\mathbf{y}})$ , are equivalent to RLS estimates when using the original data,  $(\mathbf{X}, \mathbf{y})$ ; that is,

$$\hat{\boldsymbol{\beta}}_{\text{ris}}(\alpha) = (\hat{\mathbf{X}}^T \hat{\mathbf{X}})^{-1} \hat{\mathbf{X}}^T \hat{\mathbf{y}} = (\mathbf{X}^T \mathbf{X} + \alpha \mathbf{I}_{PP})^{-1} \mathbf{X}^T \mathbf{y}. \quad (14)$$

Because the augmentation terms ( $\sqrt{\alpha} \mathbf{I}_{PP}$  and  $\mathbf{0}_P$ ) are independent of the data ( $\mathbf{X}$  and  $\mathbf{y}$ ), the iterative methods already developed for least squares (§4.1) can be used with the augmented data. Also note that the maximal eigenvalue is easily updated,  $\hat{\lambda}_{\max} = \lambda_{\max} + \alpha$  and so a new step size  $\hat{\delta}$  can be chosen without additional computation.

### 4.5 Theoretical parameter requirements for the Fan and Vercauteren scheme

We provide results to guide the choice of cryptographic parameters for the encryption scheme of Fan and Vercauteren (2012) — hereinafter FV. This is implemented in the `HomomorphicEncryptionR` package (Aslett et al., 2015a) and used in the examples.

FV represents data as a polynomial of fixed degree with coefficients in a finite integer ring.<sup>2</sup> For example,  $m$  is represented by  $\mathring{m}(x) = \sum a_i x^i$  where  $a_i$  are the binary decomposition of  $m$ , such that  $\mathring{m}(2) = m$ . Addition and multiplication operations on the ciphertext result in polynomial addition and multiplication on this representation.

The transformed regression coefficients grow substantially during computation, so that we must ensure (i) the maximal degree of the FV message polynomial is large enough to decrypt  $\tilde{\boldsymbol{\beta}}^{[K]}$ ; and (ii) that the coefficient ring is large enough to accommodate the worst case growth in coefficients.

**Lemma 3** (FV parameter requirements for GD). *If data is represented in binary decomposed polynomial form, then after running the ELS-GD algorithm the degree and coefficient value of the encrypted regression coefficients is bound by:*

$$\deg(\tilde{\boldsymbol{\beta}}^{[k]}) \leq \max\{4n + \deg(\tilde{\boldsymbol{\beta}}^{[k-1]}), (4k - 1)n\}$$

where  $\deg(\boldsymbol{\beta}^{[1]}) \leq 3n$  and  $n \equiv (\phi + 1) \log_2(10)$ ;

$$\text{and } \|\tilde{\boldsymbol{\beta}}^{[k]}\|_{\infty} \leq (4n + (n + 1)^2)NP \|\tilde{\boldsymbol{\beta}}^{[k-1]}\|_{\infty} + (4k - 3)n(n + 1)N$$

where  $\|\tilde{\boldsymbol{\beta}}^{[1]}\|_{\infty} \leq n(n + 1)N$

This lemma is proved in the supplementary materials (§2) and provides lower bounds on the choice of parameters  $d$  and  $t$  in the FV scheme.

Recall the MMD for GD is  $2K$ . Theoretical bounds on security level (Lindner and Peikert, 2011) and multiplicative depth (Lepoint and Naehrig, 2014) — together with the polynomial bounds and algorithmic MMD proved here — then enable full selection of encryption parameters to guarantee security and computational correctness of the encrypted GD algorithm.

## 5 Acceleration

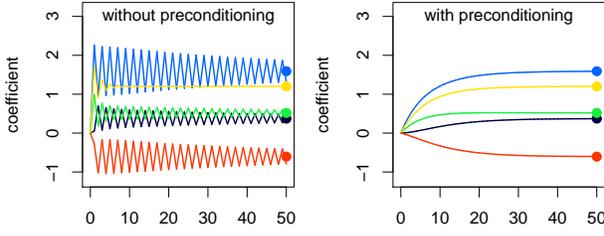
Although ELS-GD is guaranteed to converge to the OLS solution, the rate of convergence can be slow, for instance when predictors are highly correlated. Here we analyse some classic acceleration methods.

### 5.1 Preconditioning

A preconditioning matrix  $\mathbf{D}$  is often used to accelerate convergence of iterative methods (Björck, 1996, chapter 7) by solving the preconditioned problem

$$\mathbf{D}^{-1}(\mathbf{X}^T \mathbf{X} \boldsymbol{\beta} - \mathbf{X}^T \mathbf{y}) = 0 \quad (15)$$

<sup>2</sup>The maximal degree and maximal ring element are tunable parameters, but even small increases make ciphertexts bigger and homomorphic operations slower.



**Figure 1:** The effect of diagonal scaling preconditioning on the convergence paths of ELS-GD. Full circles represent OLS estimates.  $[N = 100, P = 5, \rho = 0.1]$

with the same solution as the original problem, but having more favourable spectral properties. A simple preconditioning is diagonal scaling of the columns of  $\mathbf{X}$ . Let  $\mathbf{D} = \text{diag}(d_1, \dots, d_P)$  be a  $P \times P$  diagonal matrix with diagonal entries  $\{d_1, \dots, d_P\}$  where  $d_j = \|\mathbf{X}_{\cdot j}\|_2^2$ . Since  $d_j \approx N$  for all  $j \in \mathbb{N}_{1:P}$  as a result of standardisation (see §3.1) the preconditioning matrix becomes  $\mathbf{D}^{-1} \approx N^{-1} \mathbf{I}_{PP}$ . The preconditioned update equation is then

$$\boldsymbol{\beta}^{[k]} = \boldsymbol{\beta}^{[k-1]} + (\delta/N) \mathbf{X}^T (\mathbf{y} - \mathbf{X} \boldsymbol{\beta}^{[k-1]}) \quad (16)$$

which differs from (8) only in the step size. Preconditioning smooths the convergence path, but the number of iterations required is still large (Figure 1).

## 5.2 van Wijngaarden transformation

The van Wijngaarden (1965) transformation (VWT) is a variant of the Euler transform for accelerating the convergence of an alternating series. Given the partial sums of an alternating series,  $S_{1,k} = \sum_{n=1}^k (-1)^n a_n$ , we can compute averages of these partial sums,

$$S_{z,k} = \frac{S_{z-1,k} + S_{z-1,k+1}}{2}, \quad z \in \mathbb{N}_{2:K} \quad (17)$$

(and averages of the averages) to form a matrix of averaged partial sums. For a finite number of terms ( $K$ ), averages of partial sums are often closer to the limiting value,  $S_{1,\infty}$ , than any single partial sum of the original series,  $S_{1,k}$  ( $k \in \mathbb{N}_{1:K}$ ), so that the alternating nature of the sequence is averaged out, damping its oscillatory behaviour and speeding-up convergence.

As shown in (11), the values  $\{\boldsymbol{\beta}^{[k]}\}_{k \in \mathbb{N}_{1:K}}$  computed with ELS-GD form an alternating series, making the VWT a candidate for accelerating the convergence to the true regression coefficients.

The implementation of (17) has a simple, closed form solution. For a stopping column  $k^* = \lfloor K/3 \rfloor + 1$  (as van Wijngaarden suggests) we compute the average partial sum  $S_{\star} = (2^{K-k^*})^{-1} \sum_{n=k^*}^K \binom{K-k^*}{n-k^*} S_{1,n}$ , and

take this as our best approximation to the value of the series at convergence. Notably, this can be implemented homomorphically with the exception of the division, but since the factor is independent of the data, we can compute instead  $\tilde{S}_{\star} = 2^{K-k^*} S_{\star}$  and incorporate the appropriate correction upon decryption; i.e., once ELS-GD is completed, the final VWT estimate is:

$$\tilde{\boldsymbol{\beta}}_{\text{vwt}} \leftarrow \sum_{k=k^*}^K \binom{K-k^*}{k-k^*} \tilde{\boldsymbol{\beta}}^{[k]}. \quad (18)$$

The computational cost of the VWT is minimal, involving approximately  $2K/3$  additions and multiplications, and increasing the MMD by only 1.

## 5.3 Nesterov's accelerated gradient

Nesterov's accelerated gradient (NAG) achieves a convergence rate of  $\mathcal{O}(1/K^2)$  as opposed to the  $\mathcal{O}(1/K)$  achieved by regular GD (Nesterov, 1983). The NAG algorithm can be written as follows:

$$\mathbf{s}^{[k]} = \boldsymbol{\beta}^{[k-1]} + \delta \mathbf{X}^T (\mathbf{y} - \mathbf{X} \tilde{\boldsymbol{\beta}}^{[k-1]}) \quad (19a)$$

$$\boldsymbol{\beta}^{[k]} = \mathbf{s}^{[k]} + \eta_k (\mathbf{s}^{[k]} - \mathbf{s}^{[k-1]}), \quad \eta_k < 0. \quad (19b)$$

The first step in NAG, (19a), is the same as the standard GD step in (8). The extra step, (19b), is proportional to the momentum term  $\mathbf{s}^{[k]} - \mathbf{s}^{[k-1]}$ , and is responsible for the acceleration.

These equations must also be rescaled for homomorphic computation (similarly to GD in §4.1.2):

$$\begin{aligned} \tilde{\mathbf{s}}^{[k]} &\equiv 10^{\phi} \tilde{\nu} \tilde{\boldsymbol{\beta}}^{[k-1]} + \tilde{\mathbf{X}}^T (10^{(2k-1)\phi} \tilde{\nu}^{k-1} \tilde{\mathbf{y}} - \tilde{\mathbf{X}} \tilde{\boldsymbol{\beta}}^{[k-1]}) \\ &\equiv 10^{3k\phi} \nu^k \mathbf{s}^{[k]} \end{aligned} \quad (20a)$$

$$\begin{aligned} \tilde{\boldsymbol{\beta}}^{[k]} &\equiv (10^{\phi} + \tilde{\eta}_k) \tilde{\mathbf{s}}^{[k]} - 10^{2\phi} \tilde{\nu} \tilde{\eta}_k \tilde{\mathbf{s}}^{[k-1]} \\ &\equiv 10^{(3k+1)\phi} \nu^k \boldsymbol{\beta}^{[k]} \end{aligned} \quad (20b)$$

where  $\tilde{\mathbf{y}} = 10^{\phi} \mathbf{y}$  is the transformed vector of responses, according to §3.1, and similarly for the remaining variables, except  $\{\tilde{\mathbf{s}}^{[k]}, \tilde{\boldsymbol{\beta}}^{[k]}\}_{k \in \mathbb{N}_{0:K}}$  which have iteration dependent scaling factors.

All scaling constants are independent of the data and known a priori, and so can be incorporated into the scaling by the secret key holder to obtain the final parameter estimates as:  $\text{Dec}(k_s, \tilde{\boldsymbol{\beta}}^{[K]}) / (10^{(3K+1)\phi} \nu^K)$ . Because of the extra acceleration step, ELS-NAG has a MMD equal to  $3K$  (see Table 1). This is particularly interesting, because although Nesterov's method is state-of-the-art for unencrypted GD, the increase in MMD makes it costly for encrypted analysis.

## 6 Results

In this section we empirically analyse the methods proposed for encrypted linear regression using simulated

**Table 1:** Maximum Multiplicative Depth (MMD).

Algorithm	MMD
Preconditioned gradient descent	2K
van Wijngaarden transformation	2K+1
Nesterov’s accelerated gradient	3K

and real data (see supplementary materials, §4, for details). We use the implementation of the Fan and Vercauteren (2012) cryptosystem provided by Aslett, Esperança, and Holmes (2015a). The runtimes reported for encrypted analysis are on a 48-core server. All error norms are root mean squared deviations w.r.t. OLS, and we use  $\phi = 2$  throughout.

**6.1 Simulations**

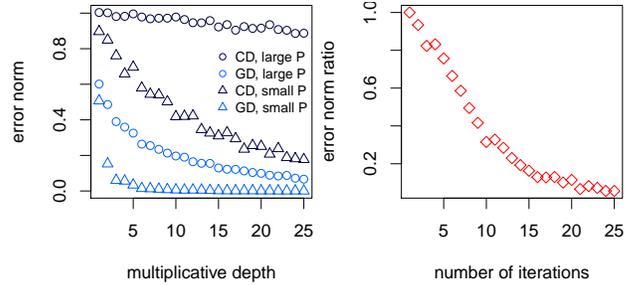
For simulations with *independent data* we generate  $\beta \sim \mathcal{N}(\mathbf{0}_p, \mathbf{I}_{pp})$ ,  $\mathbf{X} \sim \mathcal{N}(\mathbf{0}_p, \Sigma)$  and  $\mathbf{y} \sim \mathcal{N}(\mathbf{X}\beta, \mathbf{I}_{pp})$ . For simulations with *correlated data* we use Normal copulas and generate predictors whose pairwise correlations are all equal to  $\rho$ .

Figure 2 (left) illustrates the computational properties of the coordinate and gradient descent methods for encrypted regression for a fixed MMD supported in the encryption scheme is a prime determinant of the computational cost of homomorphic operations, this serves as a proxy for the error as a function of encrypted computational complexity. ELS-GD clearly outperforms ELS-CD for a fixed encrypted computational cost, as expected from §5. Furthermore, Figure 2, (right) shows the VWT provides additional acceleration in convergence relative to GD.

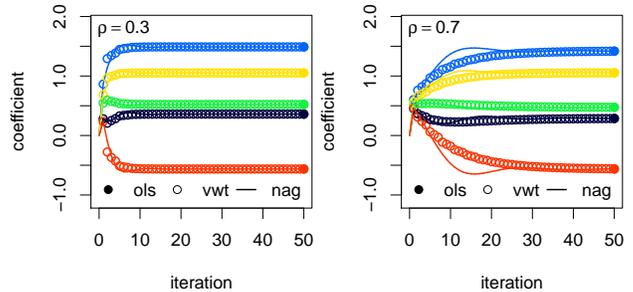
In general, higher correlation among predictors implies less favourable spectral properties for  $\mathbf{X}^T \mathbf{X}$ , which in turn makes convergence slower for both ELS-GD-VWT and ELS-NAG (Figure 3). A fair comparison must control for the fact that the two algorithms have different encrypted computational complexities. Using the MMD as a proxy again, ELS-GD-VWT typically outperforms ELS-NAG for fixed level of complexity (Figure 4). In very high correlation settings this relationship can be reversed, but only for large numbers of iterations, which it is desirable to avoid.

We stress that this choice is conditional on the encrypted computational framework considered here. It is particularly interesting that when working unencrypted, NAG is the state-of-the-art; but in the restricted framework of FHE, VWT empirically appears to be a better choice.

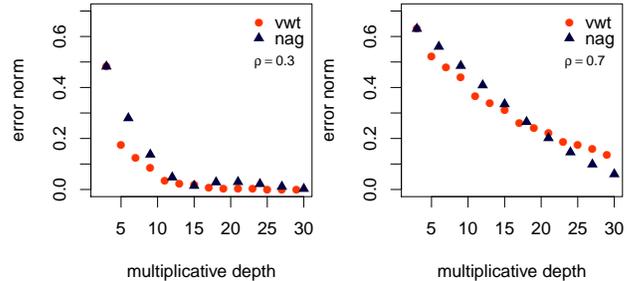
Convergence is affected by the number of predictors. In particular, the maximum number of iterations required to reduce the norm of the initial error vector by a factor  $e$  (reciprocal of the average convergence



**Figure 2:** Error norms with respect to the OLS solution. [left] ELS-CD vs. ELS-GD. Comparison for fixed multiplicative depth [right] acceleration via VWT. Ratios of error norm of ELS-GD-VWT to ELS-GD. [ $N = 100$ , large  $P = 50$ , small  $P = 5$ ]



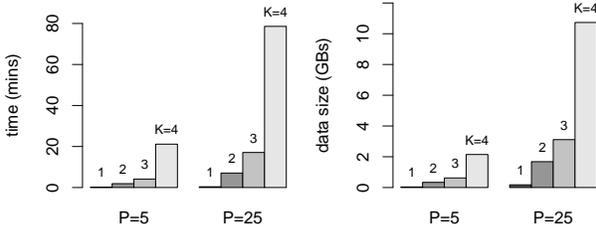
**Figure 3:** Convergence of ELS-GD-VWT and ELS-NAG for different levels of correlation: [left]  $\rho = 0.3$  [right]  $\rho = 0.7$ . [ $N = 100, P = 5$ ]



**Figure 4:** Error norm of ELS-GD-VWT and ELS-NAG as a function of multiplicative depth, for different levels of correlation: [left]  $\rho = 0.3$  [right]  $\rho = 0.7$ . [ $N = 100, P = 5$ ]

rate; Varga, 2000, p.69) gives us an idea of the relationship between number of predictors and speed of convergence. For any level of correlation, this measure of complexity increases linearly with  $P$  (see Figure 1 in the supplementary materials).

Finally, the computational costs of ELS-GD are given in Figure 5. Runtime grows quickly with the algorithm’s multiplicative depth, which increases with the number of iterations. However, for a fixed multiplicative depth the runtime is roughly linear in both  $N$  and  $P$ . Memory requirements grow in a similar fashion.



**Figure 5:** Computational aspects of ELS-GD for different problem sizes  $P \in \{2, 25\}$ , per 100 observations: [left] runtimes (in minutes) [right] encrypted data size in memory,  $\{\mathbf{X}, \mathbf{y}\}$  (in gigabytes, excluding computational overheads).

## 6.2 Applications

**Mood stability data** The first application is to mood stability in bipolar patients (Bonsall et al., 2012). Of interest in this application is the characterisation of the stochastic process governing the resulting time series, pre and post treatment, which we model as an autoregressive process of order two ( $N = 28, P = 2$ ). Convergence is achieved within 2 iterations ( $\|\beta^{[2]}\|_\infty \leq 0.04$ ; Figure 6). The algorithm runs encrypted in 12 seconds and requires under 15 MBs of memory, excluding overheads.

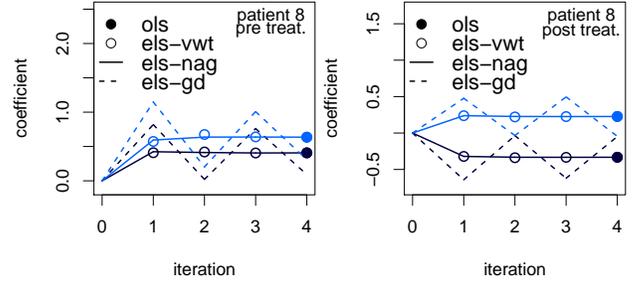
**Prostate cancer data** The second application is to prostate cancer (Stamey et al., 1989). The model here is a standard linear regression ( $N = 97, P = 8$ ). Although not all parameters have completely converged by iteration 4 with unregularised ELS-GD-VWT ( $\|\beta^{[4]}\|_\infty \leq 0.26$ ; Figure 7), the predictions are close to those produced by RLS (Figure 8). The algorithm runs encrypted in 30 minutes and requires 3.5 GBs of memory ( $K = 4$ ).

For runtimes and memory requirements in these applications see Figure 2 in the supplementary materials.

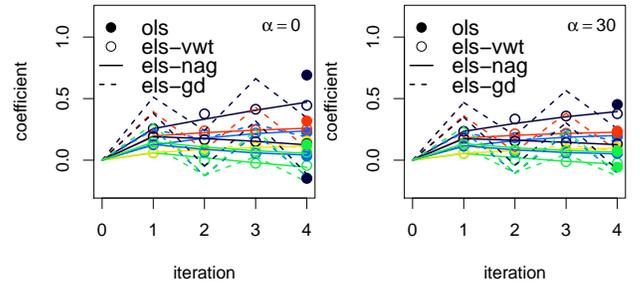
## 7 Discussion

We demonstrated that in the restricted framework of FHE, traditional state-of-the-art methods can perform poorly. Statistical and computational methods tailored for homomorphic computation are therefore required, which may differ from the state-of-the-art in an unrestricted framework.

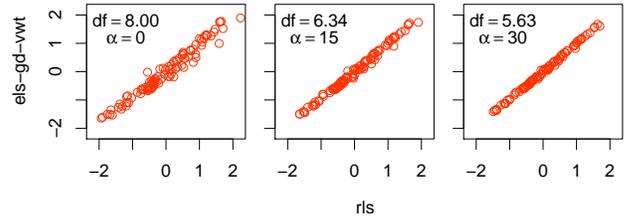
For optimal convergence speed, the step size  $\delta$  can be provided by the data holder, who can use the inequality  $\mathcal{S}(\mathbf{X}^T \mathbf{X}) \leq \|(\mathbf{X}^T \mathbf{X})^m\|^{1/m} \equiv B(m)$  to approximate  $\mathcal{S}(\mathbf{X}^T \mathbf{X})$  to arbitrary precision, since  $B(m) \rightarrow \mathcal{S}(\mathbf{X}^T \mathbf{X})$  as  $m \rightarrow \infty$ .



**Figure 6:** Convergence of different algorithms in the mood stability application (patient 8 shown): [left] pre treatment [right] post treatment. Lines of different colours represent different regression coefficients.



**Figure 7:** Convergence of different algorithms in the prostate data application: [left] without regularisation ( $\alpha = 0$ ) [right] with regularisation ( $\alpha = 30$ ).



**Figure 8:** Predictions for the prostate dataset under different regularisation settings,  $\alpha \in \{0, 15, 30\}$ .  $df = \text{trace}(\mathbf{X}(\mathbf{X}^T \mathbf{X} + \alpha \mathbf{I}_{PP})^{-1} \mathbf{X}^T)$ , effective degrees of freedom. [ $K = 4$ ]

Choosing the penalty  $\alpha$  is less straightforward. Traditional methods involve cross-validation which is impossible under strict FHE. Alternatively, it is possible to do rounds of communication and decryption between two parties to achieve this, in which case Differential Privacy can be used as a way to guarantee security during the intermediate communication steps.

## Acknowledgements

P.M. Esperança: LSI-DTC doctoral studentship (EPSRC grant EP/F500394/1). L.J.M. Aslett and C.C. Holmes: i-like project (EPSRC grant EP/K014463/1).

## References

- D. Allen. The relationship between variable selection and data augmentation and a method for prediction. *Technometrics*, 16(1):125–127, 1974.
- L. J. M. Aslett, P. M. Esperança, and C. C. Holmes. A review of homomorphic encryption and software tools for encrypted statistical machine learning. arXiv:1508.06574: arxiv.org/abs/1508.06574, 2015a.
- L. J. M. Aslett, P. M. Esperança, and C. C. Holmes. Encrypted statistical machine learning: new privacy preserving methods. arXiv:1508.06845: arxiv.org/abs/1508.06845, 2015b.
- J. Basilakis, B. Javadi, and A. Maeder. The potential for machine learning analysis over encrypted data in cloud-based clinical decision support — background and review. In *Health Informatics and Knowledge Management (HIKM'15)*, volume 164 of *Conferences in Research and Practice in Information Technology*, pages 3–13. ACS, 2015.
- A. Björck. *Numerical methods for least squares problems*. SIAM, 1996.
- M. Bonsall, S. Wallace-Hadrill, J. Geddes, G. Goodwin, and E. Holmes. Nonlinear time-series approaches in characterizing mood stability and mood instability in bipolar disorder. *Proceedings of the Royal Society B (Biological Sciences)*, 279(1730): 916–924, 2012.
- N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. Technical Report MSR-TR-2016-3.: www.microsoft.com/en-us/research/publication/cryptonets-applying-neural-networks-to-encrypted-data-with-high-throughput-and-accuracy/, 2016.
- C. Dwork and A. Roth. *The Algorithmic Foundations of Differential Privacy*. Foundations and Trends in Theoretical Computer Science. Now Publisher, 2014.
- J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. IACR Cryptology ePrint Archive, Report 2012/144: eprint.iacr.org/2012/144, 2012.
- W. Fu. Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416, 1998.
- C. Gentry. Computing arbitrary functions of encrypted data. *Communications of the ACM*, 53(3): 97–105, 2010.
- T. Graepel, K. Lauter, and M. Naehrig. ML Confidential: Machine learning on encrypted data. In *Information Security and Cryptology (ICISC'12)*, volume 7839 of *Lecture Notes in Computer Science*, pages 1–21. Springer, 2013.
- M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, and Y. Erlich. Identifying personal genomes by surname inference. *Science*, 339(6117):321–324, 2013.
- R. Hall, S. E. Fienberg, and Y. Nardi. Secure multiple linear regression based on homomorphic encryption. *Journal of Official Statistics*, 27(4):669–691, 2011.
- T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2009.
- A. Hoerl and R. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *The Journal, Technometrics*(12):1, 1970.
- T. Lepoint and M. Naehrig. A comparison of the homomorphic encryption schemes FV and YASHE. In *Progress in Cryptology (AFRICACRYPT'14)*, volume 8469 of *Lecture Notes in Computer Science*, pages 318–335. Springer, 2014.
- R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In *Topics in Cryptology (CT-RSA'11)*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer, 2011.
- P. J. McLaren, J. L. Raisaro, M. Aouri, M. Rotger, E. Ayday, I. Bartha, M. B. Delgado, Y. Vallet, H. F. Günthard, M. Cavassini, H. Furrer, T. Doco-Lecompte, C. Marzolini, P. Schmid, C. Di Benedetto, L. A. Decosterd, J. Fellay, J. Hubaux, A. Telenti, and the Swiss HIV Cohort Study. Privacy-preserving genomic testing in the clinic: a model using HIV treatment. *Genetics in Medicine*, 2016. Advance online publication.
- M. Naveed, E. Ayday, E. W. Clayton, J. Fellay, C. A. Gunter, J.-P. Hubaux, B. A. Malin, and X. Wang. Privacy in the genomic era. *ACM Computing Surveys*, 48(1):Article 6, 2015.
- Y. Nesterov. A method for solving a convex programming problem with convergence rate  $O(1/k^2)$ . *Soviet Mathematics Doklady*, 27(2):372–376, 1983.
- Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.
- V. S. Ryaben'kii and S. V. Tsynkov. *A Theoretical Introduction to Numerical Analysis*. Chapman and Hall/CRC, 2006.
- S. C. Sanderson, M. D. Linderman, S. A. Suckiel, G. A. Diaz, R. E. Zinberg, K. Ferryman, M. Wasserstein,

- A. Kasarskis, and E. E. Schadt. Motivations, concerns and preferences of personal genome sequencing research participants: baseline findings from the HealthSeq project. *European Journal of Human Genetics*, 24:14–20, 2015.
- T. Stamey, J. Kabalin, J. McNeal, I. Johnstone, F. Freiha, E. Redwine, and N. Yang. Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. II. radical prostatectomy treated patients. *Journal of Urology*, 141(5): 1076–1083, 1989.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.
- United States Executive Office of the President. Big data: Seizing opportunities, preserving values, 2014.
- A. van Wijngaarden. In *Cursus: Wetenschappelijk Rekenen B: Process Analyse*, pages 51–60. Stichting Mathematisch Centrum, 1965.
- R. S. Varga. *Matrix iterative analysis*, volume 27 of *Springer Series in Computational Mathematics*. Springer, 2nd edition, 2000.
- S. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.
- D. Wu and J. Haven. Using homomorphic encryption for large scale statistical analysis. Technical Report: [cs.stanford.edu/people/dwu4/papers/FHE-SI.Report.pdf](http://cs.stanford.edu/people/dwu4/papers/FHE-SI.Report.pdf), 2012.
- H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Methodological)*, 67(2): 301–320, 2005.