Alma Mater Studiorum
Università di Bologna

DOTTORATO DI RICERCA IN
SCIENZE STATISTICHE

CICLO XXXII

Settore Concorsuale:	13/D1
Settore Scientifico Disciplinare:	SECS-S/01

Supervised Classification with Matrix Sketching

Presentata da: Roberta Falcone

Coordinatrice Dottorato:
Prof.ssa Alessandra Luati

Supervisor:
Prof.ssa Angela Montanari

Esame Finale Anno 2020

Contents

Introduction

The recent availability of the so-called "tall" data sets, that is data sets which include information on a very large number of units, poses new challenges to statistical analysis. In particular, for many practical applications, the computational load may become incredibly high, thus making analysis very slow if not impossible.
Subsampling and divide-and-conquer approaches (i.e. divide the data in chunks, which are separately analyzed, and merge the results in a clever way) are the most commonly used approaches in this context. The quality of the approximation they yield is however hard to determine.

Data compression techniques based on random linear combinations have been developed within the computer science community and have recently become popular. They are commonly known as sketching methods. Their properties have been widely studied from a numerical analysis perspective and there is an increasing interest in assessing their behavior from a statistical perspective.
The reason for their increased popularity, besides the computational advantage that motivated their proposal, resides in the fact that they are well suited to deal with streaming data as they do not require to store the new data but can update the existing ones incrementally.
Moreover, sketching methods have also been considered under the aspect of preserving privacy (Blocki *et al.*, 2012). As the new observations are random linear combinations of the original ones, no observation in the resulting matrix can be identified with one of the original data points.

As previously mentioned, the basic idea of sketching is to reduce the size of the data set from n to k, with k much smaller than n, by creating new synthetic units obtained through random linear combinations of the units in the original data set.
The theoretical motivation for this is given by the now famous John-

son and Lindenstrauss Lemma, which gives theoretical conditions under which sketching preserves most of the linear structure that is present in the data.
This explains the wide interest that sketching has gained in the linear statistical modeling context where, for very large n, the computation of the involved Gram matrix becomes increasingly problematic. The pioneering works in this area have adopted an algorithmic perspective aimed at showing that, when the sketches are constructed appropriately, one can obtain answers that are approximately as good as the exact answer for the input data at hand, in less time than would be required to compute an exact answer.
The methods have recently attracted the interest of the statistical community with the aim of understanding whether the insights from the algorithmic perspective of sketching carry over to the statistical setting.

In the literature, two general categories of distributions for the random projection matrix have been introduced: data aware and data oblivious ones. Our focus will be on the latter approach, which will be thoroughly dealt with in the next chapters.

Data aware random projections use information from the source data to generate the random projection matrix, for instance they perform weighted sampling with replacement from the source dataset.
Relevant examples in this context are *leverage sampling* (Mahoney, 2011) and *thinning*. Thinning has been introduced in Cerioli & Perrotta (2014) with the aim of removing uninteresting or noisy observations for the estimation of regression lines in the context of robust clustering.

The theoretical motivation for sketching and the most relevant sketching methods are critically reviewed in chapter 2. In the same chapter, the literature on sketching in multiple linear regression is summarized, putting a special emphasis on the algorithmic and statistical aspects.

The goal of this thesis is to analyze the performances of sketching methods in the context of linear discriminant analysis. This theme has never been addressed before in the literature on sketching.

In chapter 1, the basic ideas of two group linear discriminant analysis are briefly summarized and a few results relating the optimal linear discriminant direction and the vector of multiple linear regression coefficients are derived (see McLachlan, 1992; Anderson, 1958). Building upon them, in chapter 3, the use of sketching methods (and in particular of what is called partial sketching) in linear discriminant analysis is evaluated according to both a numeric analytic and a statistical perspective. The performance of linear discriminant analysis on the sketched data is compared to the one obtained on the original data in a number of real datasets. In chapter 4, the properties of the sketched data are also used to solve the up to date problem of imbalanced classes in supervised classification.

To conclude it is worth mentioning that while the idea of compressing the data through random linear combinations of units is rather new and not yet fully explored, the use of random linear combinations of the columns of the data matrix, i.e. of the variables, has received a great attention.

The methods are summarized under the heading "random projections" and are aimed at performing dimension reduction, thus solving the so called "large *p* small *n*" problem.

Random projections have successfully been applied in the context of supervised classification (Cannings & Samworth, 2017), high dimensional covariance estimation (Marzetta *et al.*, 2011), clustering (Fern & Brodley, 2003), sparse principal component analysis (Gataric *et al.*, 2017), multiple linear regression (Thanei *et al.*, 2017) among others.

Chapter 1

Linear Discriminant Analysis

Linear discriminant analysis (LDA) also known as Fisher's linear discriminant analysis or as Canonical variate analysis is a widely used method aimed at finding linear combinations of observed features which characterize or separate two or more classes of objects or events. The resulting combinations are commonly used for dimensionality reduction, before later classification.

1.1 Discrimination

Let us assume we have G groups of units (each composed of n_g units, for $g = 1, \ldots, G$, such that $\sum_{g=1}^{G} n_g = n$) on which a vector random variable $\mathbf{x}$ (corresponding to p observed numeric variables) has been observed. We also assume that, for the G populations the G groups come from, the homoscedasticity condition holds i.e.

$$\Sigma_1 = \Sigma_2 = \cdots = \Sigma_g = \cdots = \Sigma_G = \Sigma.$$

Fisher (1936) suggested to look for the linear combination z of the variables in $\mathbf{x}$, $z = \ell^T \mathbf{x}$, which best separates the groups. This amounts to look, according to Fisher's perspective, for the vector ℓ such that, when projected along it, the groups are as separated as possible and as homogeneous as possible at the same time.
In this framework the function which must be optimized with respect to ℓ is the ratio of the **between group** to the **within group** variance of the linear combination z. In the $\mathbf{x}$ space the overall average is $\bar{\mathbf{x}}$, while each

group has average vector $\bar{\mathbf{x}}_g$ and covariance matrix $\mathbf{S}_g$; because of the properties of the arithmetic mean it will be

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{g=1}^{G} \bar{\mathbf{x}}_g n_g \tag{1.1}$$

The variable z will therefore have overall average $\bar{z} = \ell^T \bar{\mathbf{x}}$ and, for each group, average value $\bar{z}_g = \ell^T \bar{\mathbf{x}}_g$ and variance $Var(z_g) = \ell^T \mathbf{S}_g \ell$.

$$\begin{aligned} Var(z)_{within} &= \frac{1}{n-G}\sum_{g=1}^{G}(n_g-1)Var(z_g) = \frac{1}{n-G}\sum_{g=1}^{G}(n_g-1)\ell^T\mathbf{S}_g\ell \\ &= \ell^T\left\{\frac{1}{n-G}\sum_{g=1}^{G}(n_g-1)\mathbf{S}_g\right\}\ell = \ell^T\mathbf{W}\ell \end{aligned}$$

where $\mathbf{W} = \frac{1}{n-G}\sum_{g=1}^{G}(n_g-1)\mathbf{S}_g$ is the within group covariance matrix (also known as within group scatter matrix) in the observed variable space (it is meaningful because of the homoscedasticity assumption).

$$\begin{aligned} Var(z)_{between} &= \frac{1}{G-1}\sum_{g=1}^{G}(\bar{z}_g-\bar{z})^2 n_g = \frac{1}{G-1}\sum_{g=1}^{G}(\ell^T\bar{\mathbf{x}}_g-\ell^T\bar{\mathbf{x}})^2 n_g \\ &= \frac{1}{G-1}\ell^T\left\{\sum_{g=1}^{G} n_g(\bar{\mathbf{x}}_g-\bar{\mathbf{x}})(\bar{\mathbf{x}}_g-\bar{\mathbf{x}})^T\right\}\ell = \ell^T\mathbf{B}\ell \end{aligned}$$

where $\mathbf{B} = \frac{1}{G-1}\sum_{g=1}^{G} n_g(\bar{\mathbf{x}}_g-\bar{\mathbf{x}})(\bar{\mathbf{x}}_g-\bar{\mathbf{x}})^T$ is the between group covariance matrix (also known as between group scatter matrix) in the observed variable space. Its rank is at most $G-1$.

In the simple two group case, the between group variance has one degree of freedom only and it has the simple expression $\sum_{g=1}^{2} n_g(\bar{\mathbf{x}}_g - \bar{\mathbf{x}})(\bar{\mathbf{x}}_g-\bar{\mathbf{x}})^T$. After writing $\bar{\mathbf{x}}$ as in equation (1.1) and after little algebra, it becomes

$$\mathbf{B} = \frac{n_1 n_0}{n_1+n_0}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)^T \tag{1.2}$$

This clearly shows that in the two group case the between group covariance matrix has rank equal to 1.

According to Fisher, the function that has to be optimized with respect to ℓ is therefore:

$$\phi = \frac{Var(z)_{between}}{Var(z)_{within}} = \frac{\ell^T \mathbf{B} \ell}{\ell^T \mathbf{W} \ell} \tag{1.3}$$

In order to find the vector ℓ for which ϕ is maximum, ϕ must be derived with respect to ℓ and the derivatives must be set to 0:

$$\begin{aligned}\frac{\partial \phi}{\partial \ell} &= 2\left\{\frac{\mathbf{B}\ell(\ell^T\mathbf{W}\ell) - \mathbf{W}\ell(\ell^T\mathbf{B}\ell)}{(\ell^T\mathbf{W}\ell)^2}\right\} \\ &= 2\left\{\frac{\mathbf{B}\ell(\ell^T\mathbf{W}\ell)}{(\ell^T\mathbf{W}\ell)^2} - \frac{\mathbf{W}\ell(\ell^T\mathbf{B}\ell)}{(\ell^T\mathbf{W}\ell)^2}\right\} = 0\end{aligned}$$

Following equation (1.3), it becomes

$$\frac{\mathbf{B}\ell}{\ell^T\mathbf{W}\ell} - \frac{\mathbf{W}\ell\phi}{\ell^T\mathbf{W}\ell} = 0$$

and then

$$\mathbf{B}\ell - \phi\mathbf{W}\ell = 0 \tag{1.4}$$

or equivalently:

$$(\mathbf{B} - \phi\mathbf{W})\ell = 0$$

After pre-multiplying both sides by $\mathbf{W^{-1}}$ (under the assumption that it is non singular), we obtain:

$$(\mathbf{W}^{-1}\mathbf{B} - \phi\mathbf{I})\ell = 0$$

This is a linear homogeneous equation system which admits non trivial solution if and only if $\det(\mathbf{W}^{-1}\mathbf{B} - \phi\mathbf{I}) = 0$; this means that ϕ is an eigenvalue of $\mathbf{W}^{-1}\mathbf{B}$ and ℓ is the corresponding eigenvector.

As ϕ is the function we want to maximize we choose the largest eigenvalue and the corresponding eigenvector as the best discriminant direction.

We can find up to $G-1$ different discriminant directions as the rank of $\mathbf{B}$ (and hence of $\mathbf{W}^{-1}\mathbf{B}$) is at most $G-1$. Each of them will have a decreasing discrimination power. In the two group case, the following result holds:

Theorem 1. *Given two groups $\mathscr{G}_0$ and $\mathscr{G}_1$ coming from two homoscedastic populations, the only linear discriminant direction ℓ is proportional to:*

$$\mathbf{a} = \mathbf{W^{-1}}(\bar{\boldsymbol{x}}_\mathbf{1} - \bar{\boldsymbol{x}}_\mathbf{0})$$

Proof. If we assume that $\mathbf{W}$ is nonsingular, then equation (1.4) can be rewritten as:

$$\begin{aligned} \phi\,\ell &= \mathbf{W^{-1}B}\,\ell \\ &= \mathbf{W^{-1}}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)\,(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^\top \ell\,\frac{n_0 n_1}{n} \\ \text{Setting: } c = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^\top \ell \in \mathbb{R} & \\ \ell &= \underbrace{\frac{c}{\phi}\,\frac{n_0 n_1}{n}}_{\in \mathbb{R}}\,\mathbf{W^{-1}}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0) \\ \text{and hence:}\quad & \ell \propto \mathbf{a} \end{aligned}$$

□

In the same paper Fisher rephrased the issue as a multiple linear regression problem.
Given $\mathbf{Y} = \mathbf{X}\beta + \varepsilon$, where $\mathbf{X}$ is mean centered and the variable $\mathbf{y}$ identifying group membership is such that:

$$y_i = \begin{cases} -n_1/n & \text{if the unit } i \in \mathscr{G}_0 \\ n_0/n & \text{if the unit } i \in \mathscr{G}_1 \end{cases}$$

the solution to the least square regression problem relating $\mathbf{Y}$ to $\mathbf{X}$, is

$$\mathbf{b} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} = (TSS)^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)\frac{n_0 n_1}{n_0 + n_1} \tag{1.5}$$

where $TSS = \mathbf{X}^\top\mathbf{X}$ is the total sum of squares.

The following proposition can be proved:

Proposition 1.1. *the best linear discriminant direction ℓ is also proportional to the vector of linear regression coefficients $\boldsymbol{b}$.*

Proof. Remembering that:

$$\mathbf{B} = \frac{BSS}{G-1}, \; \mathbf{W} = \frac{WSS}{n-G}$$

where BSS and WSS are the between and the within sum of squares, equation (1.4) can be rewritten as:

$$\frac{BSS}{G-1}\ell - \frac{WSS}{n-G}\ell\phi = 0$$

or equivalently as:

$$BSS\,\ell - WSS\,\ell\,\phi\,\frac{G-1}{n-G} = 0 \tag{1.6}$$

and hence:

$$BSS\,\ell - WSS\,\ell\,\phi_1 = 0, \quad \text{where } \phi_1 = \phi\,\frac{G-1}{n-G}$$

This means that ℓ is also an eigenvector of $WSS^{-1}BSS$ and ϕ_1 is the corresponding eigenvalue.

As: $WSS = TSS - BSS$, then (1.6) becomes:

$$\begin{aligned}
&BSS\,\ell - TSS\,\ell\,\phi_1 + BSS\,\ell\,\phi_1 = 0\\
&BSS\,\ell(1+\phi_1) - TSS\,\ell\,\phi_1 = 0\\
&BSS\,\ell - TSS\,\ell\,\frac{\phi_1}{(1+\phi_1)} = 0\\
&BSS\,\ell - TSS\,\ell\,\phi_2 = 0, \quad \text{where } \phi_2 = \frac{\phi_1}{1+\phi_1}\\
\text{So:}\quad &(TSS)^{-1}BSS\,\ell = \phi_2\,\ell
\end{aligned}$$

Therefore, the best linear discriminant direction ℓ is also an eigenvector of $TSS^{-1}BSS$ and ϕ_2 is the corresponding eigenvalue.
Moreover, in the two group case:

$$\begin{aligned}\phi_2\,\ell &= (TSS)^{-1}BSS\,\ell\\ &= (TSS)^{-1}\mathbf{B}\,\ell\\ &= (TSS)^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_\mathbf{0})\,(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_\mathbf{0})^\top\,\ell\,\frac{n_0\,n_1}{n}\end{aligned}$$

Setting $c=(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_\mathbf{0})^\top\,\ell\in\mathbb{R}$, then :

$$\begin{aligned}\ell &= \frac{c}{\phi_2}\,TSS^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_\mathbf{0})\,\frac{n_0\,n_1}{n}\\ &= \frac{c}{\phi_2}\mathbf{b}\\ \text{So:}\qquad \ell &\propto \mathbf{b}\end{aligned}$$

□

1.2 Classification based on probability models

Let $\mathbf{x}$ be the p-dimensional vector of the observed variables and $\mathbf{x}_{new}$ a new observation whose group membership is unknown. Let Π_1 and Π_0 denote the two parent populations. The key assumption is that $\mathbf{x}$ has a different probability density function (*pdf*) in Π_1 and Π_0.
Let us denote the *pdf* of $\mathbf{x}$ in Π_1 as $f_1(\mathbf{x})$ and the *pdf* of $\mathbf{x}$ in Π_0 as $f_0(\mathbf{x})$.
Let us denote by R the set of all possible values $\mathbf{x}$ can assume.
As $f_1(\mathbf{x})$ and $f_0(\mathbf{x})$ usually overlap, each point of R can belong both to Π_1 and Π_0, but with a different probability degree. The goal is to partition R into two exhaustive, non overlapping regions R_1 and R_0 ($R_1\cup R_0=R$ and $R_1\cap R_0=\emptyset$) such that the probability of a wrong classification is minimum, when a unit belonging to R_1 is allocated to Π_1 and a unit belonging to R_0 is allocated to Π_0.
Given a new unit $\mathbf{x}_{new}$, whose group membership is unknown, a very intuitive rule consists in allocating it to Π_1 if the probability that it comes from Π_1 is larger than the probability that it comes from Π_0 or to allocate it to Π_0 if the opposite holds.
According to this criterion:
R_1 is the set of the $\mathbf{x}$ values such that $f_1(\mathbf{x})>f_0(\mathbf{x})$ and R_0 is the set of the $\mathbf{x}$ values such that $f_1(\mathbf{x})<f_0(\mathbf{x})$. The ensuing classification rule is

therefore: allocate $\mathbf{x}_{new}$ to Π_1 if:

$$\frac{f_1(\mathbf{x}_{new})}{f_0(\mathbf{x}_{new})} > 1$$

allocate $\mathbf{x}_{new}$ to Π_0 if:

$$\frac{f_1(\mathbf{x}_{new})}{f_0(\mathbf{x}_{new})} < 1$$

allocate $\mathbf{x}_{new}$ randomly to one of the two populations if equality holds.
This classification rule is known as *likelihood ratio rule*.
However intuitively reasonable, this rule neglects possibly different prior probabilities of class membership and possibly different misclassification costs.
Let us denote by π_1 the prior probability that $\mathbf{x}_{new}$ belongs to Π_1 and by π_0 the prior probability that $\mathbf{x}_{new}$ belongs to Π_0 $(\pi_1 + \pi_0 = 1)$.
Based on likelihoods only, the probability that a unit belonging to Π_1 is wrongly classified to Π_0 (this happens when it falls in R_0) is:

$$p(0|1) = \int_{R_0} f_1(\mathbf{x}) d\mathbf{x}$$

and the probability of a wrong classification of a unit to Π_1 when it effectively comes from Π_0 is:

$$p(1|0) = \int_{R_1} f_0(\mathbf{x}) d\mathbf{x}$$

The overall probability of a wrong classification is therefore:

$$prob = \pi_1 \, p(0|1) + \pi_0 \, p(1|0)$$

R_1 and R_0 should be therefore chosen in such a way that *prob* is minimum.
prob can be written as

$$prob = \pi_1 \int_{R_0} f_1(\mathbf{x}) d\mathbf{x} + \pi_0 \int_{R_1} f_0(\mathbf{x}) d\mathbf{x} \tag{1.7}$$

Since R is the complete space and *pdfs* are known to integrate to 1 over their domain, it is:

$$\int_R f_1(\mathbf{x}) d\mathbf{x} = \int_R f_0(\mathbf{x}) d\mathbf{x} = 1$$

and as $R_1 \cup R_0 = R$ and $R_1 \cap R_0 = \emptyset$ we have:

$$\int_R f_1(\mathbf{x})d\mathbf{x} = \int_{R_1} f_1(\mathbf{x})d\mathbf{x} + \int_{R_0} f_1(\mathbf{x})d\mathbf{x} = 1$$

and hence:

$$\int_{R_0} f_1(\mathbf{x})d\mathbf{x} = 1 - \int_{R_1} f_1(\mathbf{x})d\mathbf{x}$$

After replacing it in the equation (1.7) we obtain:

$$\begin{aligned} prob &= \pi_1 \left[1 - \int_{R_1} f_1(\mathbf{x})d\mathbf{x}\right] + \pi_0 \int_{R_1} f_0(\mathbf{x})d\mathbf{x} \\ &= \pi_1 - \pi_1 \int_{R_1} f_1(\mathbf{x})d\mathbf{x} + \pi_0 \int_{R_1} f_0(\mathbf{x})d\mathbf{x} \\ &= \pi_1 + \int_{R_1} \left[\pi_0 f_0(\mathbf{x}) - \pi_1 f_1(\mathbf{x})\right] d\mathbf{x} \end{aligned}$$

As π_1 is a constant, *prob* will be minimum when the integral is minimum, *i.e.* when the integrand is negative.
This means that R_1 should be chosen so that, for the points belonging to it:

$$\pi_0 f_0(\mathbf{x}) - \pi_1 f_1(\mathbf{x}) < 0$$

that is:

$$\pi_1 f_1(\mathbf{x}) > \pi_0 f_0(\mathbf{x})$$

The ensuing allocation rule will then be:
allocate $\mathbf{x}_{new}$ to Π_1 if:

$$\frac{f_1(\mathbf{x}_{new})}{f_0(\mathbf{x}_{new})} > \frac{\pi_0}{\pi_1}$$

allocate $\mathbf{x}_{new}$ to Π_0 if:

$$\frac{f_1(\mathbf{x}_{new})}{f_0(\mathbf{x}_{new})} < \frac{\pi_0}{\pi_1}$$

allocate $\mathbf{x}_{new}$ randomly to one of the two populations if equality holds.
In the equal prior case $(\pi_1 = \pi_2 = 1/2)$ the likelihood ratio rule is obtained.
It is worth adding that the classification rule obtained by minimizing the total probability of a wrong classification is equivalent to the one that would be obtained by maximizing the posterior probability of population membership. That's the reason why it is often called an optimal

Bayes rule.

Gaussian populations

Let us assume that both $f_1(\mathbf{x})$ and $f_0(\mathbf{x})$ are multivariate normal densities with parameters (μ_1, Σ_1) and (μ_0, Σ_0), respectively:

$$f_1(\mathbf{x}) = (2\pi)^{-p/2} |\Sigma_1|^{-1/2} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \mu_1)^\top \Sigma_1^{-1} (\mathbf{x} - \mu_1) \right\}$$

$$f_0(\mathbf{x}) = (2\pi)^{-p/2} |\Sigma_0|^{-1/2} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \mu_0)^\top \Sigma_0^{-1} (\mathbf{x} - \mu_0) \right\}$$

The likelihood ratio is therefore:

$$\frac{f_1(\mathbf{x})}{f_0(\mathbf{x})} = |\Sigma_1|^{-1/2} |\Sigma_0|^{1/2} \exp\left\{ -\frac{1}{2} \left[(\mathbf{x} - \mu_1)^\top \Sigma_1^{-1} (\mathbf{x} - \mu_1) - (\mathbf{x} - \mu_0)^\top \Sigma_0^{-1} (\mathbf{x} - \mu_0) \right] \right\}$$

$$= |\Sigma_1|^{-1/2} |\Sigma_0|^{1/2} \exp\{ -\frac{1}{2} [\mathbf{x}^\top (\Sigma_1^{-1} - \Sigma_0^{-1}) \mathbf{x} - 2\mathbf{x}^\top (\Sigma_1^{-1} \mu_1 - \Sigma_0^{-1} \mu_0)$$
$$+ \mu_1^\top \Sigma_1^{-1} \mu_1 - \mu_0^\top \Sigma_0^{-1} \mu_0] \}$$

The expression can be simplified by considering the logarithm of the likelihood ratio. In this way the so called Quadratic discriminant function is obtained:

$$Q(\mathbf{x}) = \ln \frac{f_1(\mathbf{x})}{f_0(\mathbf{x})}$$
$$= \frac{1}{2} \ln \frac{|\Sigma_0|}{|\Sigma_1|} - \frac{1}{2} [\mathbf{x}^\top (\Sigma_1^{-1} - \Sigma_0^{-1}) \mathbf{x} - 2\mathbf{x}^\top (\Sigma_1^{-1} \mu_1 - \Sigma_0^{-1} \mu_0)$$
$$+ \mu_1^\top \Sigma_1^{-1} \mu_1 - \mu_0^\top \Sigma_0^{-1} \mu_0]$$

The expression clearly shows that it is a quadratic function of $\mathbf{x}$.
The ensuing classification rule suggests to allocate $\mathbf{x}_{new}$ to Π_1 if:

$$Q(\mathbf{x}_{new}) > \ln(H)$$

where H is either 1, if equal priors are assumed, or $\frac{\pi_0}{\pi_1}$ in the unequal prior case.
When $\Sigma_1 = \Sigma_0 = \Sigma$ then the likelihood ratio becomes:

$$\frac{f_1(\mathbf{x})}{f_0(\mathbf{x})} = \exp\left\{ (\mu_1 - \mu_0)^\top \Sigma^{-1} \mathbf{x} - \frac{1}{2} (\mu_1 - \mu_0)^\top \Sigma^{-1} (\mu_1 + \mu_0) \right\}$$
$$= \exp\left\{ (\mu_1 - \mu_0)^\top \Sigma^{-1} \left[\mathbf{x} - \frac{1}{2} (\mu_1 + \mu_0) \right] \right\}$$

After taking the logarithm we obtain:

$$L(\mathbf{x}) = (\mu_1 - \mu_0)^\top \Sigma^{-1} \left[\mathbf{x} - \frac{1}{2}(\mu_1 + \mu_0)\right] \tag{1.8}$$

This is known as linear discriminant rule as it is a linear function of $\mathbf{x}$. The ensuing classification rule suggests to allocate $\mathbf{x}_{new}$ to Π_1 if

$$L(\mathbf{x}_{new}) > \ln(H)$$

where H is either 1, if equal priors are assumed, or $\frac{\pi_0}{\pi_1}$ in the unequal prior case.
In empirical applications, maximum likelihood estimates of the model parameters are plugged into the classification rules.
μ_1 and μ_0 are estimated by $\bar{\mathbf{x}}_\mathbf{1}$ and $\bar{\mathbf{x}}_\mathbf{0}$. Furthermore, in the heteroscedastic case Σ_1 and Σ_0 are replaced by S_1 and S_0 respectively, while in the homoscedastic case, the common Σ is replaced by the within group covariance matrix $\mathbf{W}$.
Equation (1.8) clearly tells us that in the two group case, the linear combination minimizing the total probability of a wrong classification is given by:

$$(\mu_1 - \mu_0)^\top \Sigma^{-1} \mathbf{x}$$

Its sample version is obtained through the vector:

$$\mathbf{a} = W^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0) \tag{1.9}$$

1.3 Optimal direction and regression

We have verified that the vector ℓ of the coefficients of the linear combination maximizing the ratio of the between to the within variance is proportional to **a**.
We have also said that ℓ is proportional to the vector of the linear regression coefficients **b**.
It follows that **a** is proportional to **b**.
We can prove the following proposition:

Proposition 1.2. *The relation between* **a** *and* **b** *is given as:*

$$\mathbf{a} = \gamma\,\mathbf{b} = \gamma(\mathbf{X}^\top\mathbf{X})^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)\,\frac{n_0\,n_1}{n} \tag{1.10}$$

where:

$$\gamma = \frac{n}{n_0\,n_1}\,(n-2) + d_M^2(\bar{\mathbf{x}}_0, \bar{\mathbf{x}}_1) \tag{1.11}$$

and $d_M^2(\bar{\mathbf{x}}_0, \bar{\mathbf{x}}_1) = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^\top W^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)$ *is the squared Mahalanobis Distance between* $\bar{\mathbf{x}}_0$ *and* $\bar{\mathbf{x}}_1$.

Proof. $\mathbf{a} \propto \mathbf{b} \Rightarrow \mathbf{a} = \gamma\,\mathbf{b}$, where $\gamma \in \mathbb{R}$

$$\begin{aligned}\text{From (1.9)} \quad & \mathbf{a} = W^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0),\\ \text{from (1.10)} \quad & \mathbf{b} = (TSS)^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)\,\frac{n_0\,n_1}{n}\end{aligned}$$

So:

$$\begin{aligned}
W^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0) &= \gamma\,(TSS)^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\,\frac{n_0\,n_1}{n}\\
(n-2)(WSS)^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0) &= \gamma\,(WSS+BSS)^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\,\frac{n_0\,n_1}{n}\\
\gamma\,(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0) &= \frac{n}{n_0\,n_1}(n-2)(WSS+BSS)(WSS)^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\\
\gamma\,(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0) &= \frac{n}{n_0\,n_1}(n-2)I\,(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)+\frac{n}{n_0\,n_1}(n-2)\,BSS\,(WSS)^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\\
\gamma\,(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0) &= \frac{n}{n_0\,n_1}(n-2)(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)+(n-2)(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)^\top(WSS)^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\\
\gamma\,(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0) &= \frac{n}{n_0\,n_1}(n-2)(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)+d_M^2(\bar{\mathbf{x}}_0,\bar{\mathbf{x}}_1)\,(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\\
\gamma\,(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0) &= \left[\frac{n}{n_0\,n_1}(n-2)+d_M^2(\bar{\mathbf{x}}_0,\bar{\mathbf{x}}_1)\right](\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\\
\text{and hence:}\quad \gamma &= \frac{n}{n_0\,n_1}(n-2)+d_M^2(\bar{\mathbf{x}}_0,\bar{\mathbf{x}}_1)
\end{aligned}$$

□

Alternatively, starting from the original formulation for **a** (1.9), we can obtain:

Proposition 1.3.

$$\mathbf{a}=\frac{n}{n_0\,n_1}\,(n-2)\,\frac{1}{(1-\frac{n_0 n_1}{n}\,\delta^2)}\,\mathbf{b} \qquad (1.12)$$

where: $\delta^2=(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)^\top\,(\mathbf{X}^\top\mathbf{X})^{-1}\,(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\in\mathbb{R}$

Proof. The proof is based on a result by Miller (1981) according to which, given a full rank matrix $\mathbf{X}^\top\mathbf{X}$ and a rank 1 matrix B:

$$(\mathbf{X}^\top\mathbf{X}-B)^{-1}=(\mathbf{X}^\top\mathbf{X})^{-1}+\frac{1}{1-g}(\mathbf{X}^\top\mathbf{X})^{-1}\,B\,(\mathbf{X}^\top\mathbf{X})^{-1} \qquad (1.13)$$

where: $g=tr\left[(\mathbf{X}^\top\mathbf{X})^{-1}B\right]=\frac{n_0 n_1}{n}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)^\top\,(\mathbf{X}^\top\mathbf{X})^{-1}\,(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)=\frac{n_0 n_1}{n}\,\delta^2$

So:

$$
\begin{aligned}
\mathbf{a} &= W^{-1}\,(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\\
&= (n-2)\,(\mathbf{X}^\top\mathbf{X}-B)^{-1}\,(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\\
&= (n-2)\,\left[(\mathbf{X}^\top\mathbf{X})^{-1}+\frac{1}{1-g}(\mathbf{X}^\top\mathbf{X})^{-1}B\,(\mathbf{X}^\top\mathbf{X})^{-1}\right](\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\\
&= (n-2)\,\left[(\mathbf{X}^\top\mathbf{X})^{-1}+\frac{1}{1-g}(\mathbf{X}^\top\mathbf{X})^{-1}\,\frac{n_0n_1}{n}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)^\top(\mathbf{X}^\top\mathbf{X})^{-1}\right](\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\\
&= (n-2)\,\left[(\mathbf{X}^\top\mathbf{X})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)+\frac{1}{1-\frac{n_0n_1}{n}\,\delta^2}\,(\mathbf{X}^\top\mathbf{X})^{-1}\,\frac{n_0n_1}{n}\,(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\,\delta^2\right]\\
&= (n-2)\,\frac{1}{1-\frac{n_0n_1}{n}\,\delta^2}\,(\mathbf{X}^\top\mathbf{X})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\\
&= \frac{n}{n_0n_1}\,(n-2)\,\frac{1}{1-\frac{n_0n_1}{n}\,\delta^2}\,(\mathbf{X}^\top\mathbf{X})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\,\frac{n_0n_1}{n}\\
&= \underbrace{\frac{n}{n_0n_1}\,(n-2)\,\frac{1}{(1-\frac{n_0n_1}{n}\delta^2)}}_{\gamma^*\in\mathbb{R}}\,\mathbf{b}
\end{aligned}
$$

□

We have seen that $\mathbf{a}=\gamma\,\mathbf{b}$ and $\mathbf{a}=\gamma^*\mathbf{b}$, so $\gamma=\gamma^*$.
Starting from this equality, we can rewrite the squared Mahalanobis distance in terms of the δ^2 constant, and vice versa.
So:

$$d_M^2(\bar{\mathbf{x}}_0,\bar{\mathbf{x}}_1)=\frac{(n-2)\,\delta^2}{1-\frac{n_0n_1}{n}\,\delta^2} \tag{1.14}$$

and:

$$\delta^2=\frac{d_M^2(\bar{\mathbf{x}}_0,\bar{\mathbf{x}}_1)}{(n-2)+\frac{n_0n_1}{n}\,d_M^2(\bar{\mathbf{x}}_0,\bar{\mathbf{x}}_1)} \tag{1.15}$$

If we think of our classification problem as a regression problem, the following expression for the Model Sum of Squares (MSS) can be easily derived:

Proposition 1.4.

$$MSS=\frac{n_0n_1}{n}-\frac{(n-2)}{(n-2)\frac{n_0n_1}{n}+d_M^2(\bar{\mathbf{x}}_0,\bar{\mathbf{x}}_1)} \tag{1.16}$$

or also as:

$$MSS = \left(\frac{n_0 n_1}{n}\right)^2 \delta^2 \tag{1.17}$$

Proof.

$$\begin{aligned}
MSS &= \mathbf{b}^\top \mathbf{X}^\top \mathbf{y} \\
&= \frac{1}{\gamma}\, \mathbf{a}^\top \mathbf{X}^\top \mathbf{y} \\
&= \frac{1}{\gamma}\, (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^\top W^{-1} (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)\, \frac{n_0\, n_1}{n} \\
&= \frac{1}{\gamma}\, d_M^2(\bar{\mathbf{x}}_0, \bar{\mathbf{x}}_1)\, \frac{n_0\, n_1}{n} \\
&= \frac{1}{\frac{n}{n_0 n_1}(n-2) + d_M^2(\bar{\mathbf{x}}_0, \bar{\mathbf{x}}_1)}\, d_M^2(\bar{\mathbf{x}}_0, \bar{\mathbf{x}}_1)\, \frac{n_0\, n_1}{n} \\
&= \frac{n_0 n_1}{n} \left(1 - \frac{(n-2)\,\frac{n}{n_0\, n_1}}{(n-2)\frac{n}{n_0\ n_1} + d_M^2(\bar{\mathbf{x}}_0, \bar{\mathbf{x}}_1)}\right) \\
&= \boxed{\frac{n_0 n_1}{n} - \frac{(n-2)}{(n-2)\frac{n}{n_0\, n_1} + d_M^2(\bar{\mathbf{x}}_0, \bar{\mathbf{x}}_1)}} \\
&= \frac{n_0 n_1}{n} - \frac{(n-2)}{\gamma} \\
&= \frac{n_0 n_1}{n} - \frac{(n-2)}{\gamma^*} \\
&= \frac{n_0 n_1}{n} - \frac{(n-2)}{\frac{n}{n_0\, n_1}\,(n-2)\,\frac{1}{(1-\frac{n_0 n_1}{n}\delta^2)}} \\
&= \frac{n_0 n_1}{n} - \frac{n_0 n_1}{n}\,(1 - \frac{n_0 n_1}{n}\delta^2) \\
&= \boxed{\left(\frac{n_0 n_1}{n}\right)^2 \delta^2}
\end{aligned}$$

□

Chapter 2

Matrix Sketching

2.1 Motivation

Matrix sketching is a probabilistic data compression technique. Its goal is to reduce the number of rows in a data set and the task is accomplished by linearly combining the rows of the original data set through randomly generated coefficients. The analysis can then be performed on the reduced matrix, thus saving time and space.

The theoretical justification for this approach to data compression is given by Johnson-Lindenstrauss lemma (Johnson & Lindenstrauss, 1984).

Lemma 2.1. ***Johnson Lindenstrauss (1984)****. Let Q be a subset of $p-$points in $\mathbb{R}^n$, then for any $\varepsilon \in (0,1/2)$ and for $k = \dfrac{20logp}{\varepsilon^2}$ there exists a Lipschitz mapping $f : \mathbb{R}^n \longrightarrow \mathbb{R}^k$ such that for all $\mathbf{u}$, $\mathbf{v} \in Q$:*

$$(1-\varepsilon)||\mathbf{u}-\mathbf{v}||^2 \leq ||f(\mathbf{u})-f(\mathbf{v})||^2 \leq (1+\varepsilon)||\mathbf{u}-\mathbf{v}||^2$$

The Lemma says that any $p-$point subset of the Euclidean space can be embedded in k dimensions without distorting the distances between any pair of points by more than a factor of $1 \pm \varepsilon$, for any ε in $(0,1/2)$.

Moreover, it also gives an explicit bound on the dimensionality required for a projection to ensure that it will approximately preserve distances. This bound depends on the dimension of the data matrix that is not sketched, i.e. p in this case.

The original proof by Johnson and Lindenstrauss is probabilistic, showing that projecting the p-point subset onto a random k-dimensional subspace only changes the inter-point distances by $1 \pm \varepsilon$ with positive probability. The proof of the lemma is based on what is called *norm preservation* lemma (Dasgupta & Gupta, 2003).

Lemma 2.2. ***Norm preservation lemma****. Let* $\mathbf{x} \in \mathbb{R}^n$*. Assume that the entries of a matrix* $\mathbf{A} \subset \mathbb{R}^{k \times n}$ *are sampled independently from a* $N(0,1/k)$*, then*

$$Pr((1-\varepsilon)\|\mathbf{x}\|^2 \leq \|\mathbf{A}\mathbf{x}\|^2 \leq (1+\varepsilon)\|\mathbf{x}\|^2) \geq 1 - 2e^{-(\varepsilon^2-\varepsilon^3)k/4}$$

By applying the norm preservation lemma to the vectors $\mathbf{u}+\mathbf{v}$ and $\mathbf{u}-\mathbf{v}$ the following corollary can be proved.

Corollary 1. *Let* $\mathbf{u}$*,* $\mathbf{v} \in \mathbb{R}^n$ *and* $\|\mathbf{u}\| < 1$*,* $\|\mathbf{v}\| < 1$*. Let* $f = \mathbf{A}\mathbf{x}$ *where* $\mathbf{A}$ *is a* $k \times n$ *matrix, where each entry is sampled i.i.d from a Gaussian* $N(0,1/k))$*. Then,*

$$Pr(|\mathbf{u}^\top \mathbf{v} - f(\mathbf{u})f(\mathbf{v})| \geq \varepsilon) \leq 4e^{-(\varepsilon^2-\varepsilon^3)k/4}$$

The corollary states that inner products are preserved as well after random projections.

In order to apply Johnson and Lindenstrauss lemma, the concept of $\varepsilon-$subspace embedding is useful.

Definition 1. ***$\varepsilon-$subspace embedding.*** *For a given* $n \times p$ *matrix* $\mathbf{X}$*, we call a* $k \times n$ *random matrix* $\mathbf{S}$ *an* ε*-subspace for* $\mathbf{X}$*, if for all vectors* $\mathbf{z} \in \mathbb{R}^p$

$$(1-\varepsilon)\|\mathbf{X}\mathbf{z}\|^2 \leq \|\mathbf{S}\mathbf{X}\mathbf{z}\|^2 \leq (1+\varepsilon)\|\mathbf{X}\mathbf{z}\|^2$$

$\mathbf{S}$ is usually called a *Sketching Matrix*. It reduces the sample size from n to k whilst preserving most of the linear information in the full dataset.

2.2 Sketching Methods

The original proof by Johnson and Lindenstrauss required $\mathbf{S}$ to have orthogonal rows; subsequent proofs relaxed the orthogonality requirement and assumed the entries of $\mathbf{S}$ to be independently randomly generated from a Gaussian distribution with 0 mean and variance equal to $1/k$. This approach to sketching is known as Gaussian sketching and it is largely used in statistical applications as it allows for statistical analysis of the results obtained after sketching.

Although appealing from a theoretical point of view, Gaussian sketching is computationally demanding as the associated sketching matrix is full. Therefore research has been oriented towards developing more efficient algorithms still satisfying the ε-subspace embedding property.
Ailon & Chazelle (2009) have proposed what is known as Hadamard sketch. The sketching matrix is formed as $\mathbf{S} = \Phi\mathbf{HD}/\sqrt{k}$, where Φ is a $k \times n$ matrix and $\mathbf{H}$ and $\mathbf{D}$ are both $n \times n$ matrices. The fixed matrix $\mathbf{H}$ is a Hadamard matrix of order n. A Hadamard matrix is a square matrix with elements that are either $+1$ or -1 and orthogonal rows. Hadamard matrices do not exist for all integers n, the source dataset can be padded with zeros so that a conformable Hadamard matrix is available. The random matrix $\mathbf{D}$ is a diagonal matrix where each nonzero element is an independent Rademacher random variable. The random matrix Φ subsamples k rows of $\mathbf{H}$ with replacement. The structure of the Hadamard sketch allows for fast matrix multiplication, reducing calculation of the sketched dataset from $O(npk)$ of the gaussian sketch to $O(np\log k)$ operations.
Another efficient method for generating $\varepsilon-$subspace embeddings is the so-called Clarkson-Woodruff sketch (Clarkson & Woodruff, 2017). The sketching matrix is a sparse random matrix $\mathbf{S} =\Gamma\mathbf{D}$, where $\Gamma(k \times n)$ and $\mathbf{D}(n \times n)$ are two independent random matrices. The matrix Γ is a random matrix with only one element for each column set to +1. The matrix $\mathbf{D}$ is the same as above. This results in a sparse random matrix $\mathbf{S}$ with only one nonzero entry per column. The sparsity speeds up matrix multiplication, dropping the complexity of generating the sketched dataset to $O(np)$.

It is worth noticing that the rows of the Gaussian and Clarkson-Woodruff sketching matrices are not orthogonal and this implies that the geometry of the original space is not preserved after sketching. The

Gaussian sketching matrix is sometimes orthogonalized according to Gram-Schmidt procedure thus leading to what are known as Haar projections. This operation inevitably increases the computational load. Hadamard sketching matrices on the contrary are orthogonal by construction.

In the following we will denote by $\mathbf{X}$ the $n \times p$ original data matrix and by $\tilde{\mathbf{X}} = \mathbf{S}\mathbf{X}$ the $k \times p$ sketched data matrix.

As said in the Introduction, the leading motivation for sketching is the reduction of the computational cost related to the computation of the Gram matrix. In order to study the performances of the different sketching methods in terms of quality of the approximation of the Gram matrix after sketching, we have run a small simulation study in which a $n \times p$ data matrix has been generated for $n = \{1024, 2560\}$ (the numbers have been chosen in such a way that they are conformable to the Hadamard matrix). The matrices have been sketched 200 times and every time the Frobenius norm of the difference between the original Gram matrix $\mathbf{X}^\top\mathbf{X}$ and the sketched one $\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}}$ has been computed.

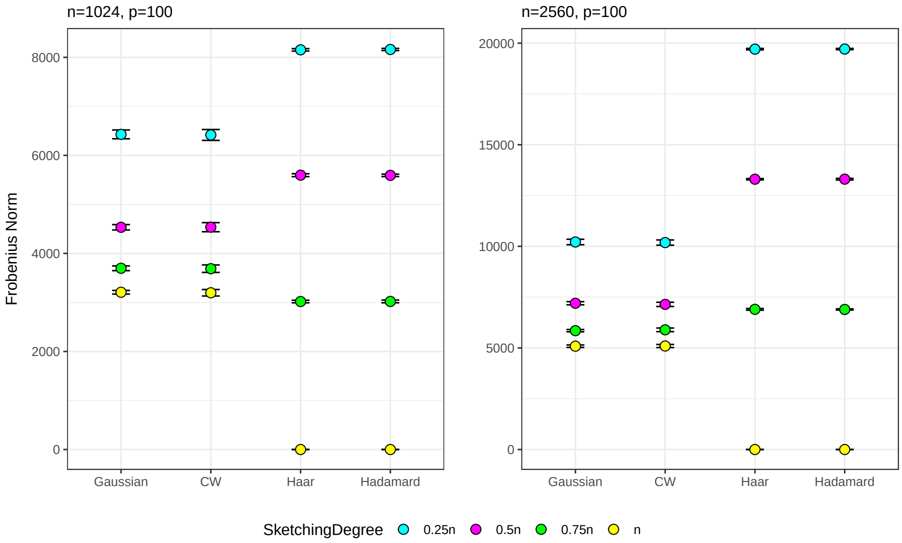

Figure 2.1: Frobenius norm of the difference between the original and the sketched Gram matrix, different sketching methods (Gaussian, Clarkson-Woodruff, Haar, Hadamard), for different sketching degrees (0.25 n, 0.5n, 0.75n, 1n) and different data set sizes (1024, 2560).

Figure 2.1 reports the boxplots of the Frobenius norm over the 200 replicates for the different sketching algorithms, different data sizes and four different degrees of sketching corresponding to $k = n$, $k = 0.75n$ $k = 0.5n$ and $k = 0.25n$. When $k = n$ the two orthogonal sketching methods (namely Haar and Hadamard) exactly reproduce the Gram matrix while, due to non-orthogonal rows, the Gaussian sketch and the Clarkson-Woodruff one give distorted results. However, especially for large datasets, when the data set size is reduced through Gaussian or Clarkson-Woodruff sketching, the approximation of the Gram matrix is better than the one obtained by the two orthogonal sketching methods. In particular, Gaussian and Clarkson-Woodruff sketching guarantee the same degree of approximation in terms of Frobenius norm but with a stronger reduction in the data set size. For this reason and for its relevant statistical properties, most of the following results will be referred to as Gaussian sketching.

2.3 Matrix Sketching in linear regression

Maybe the first context in which matrix sketching has been applied is multiple linear regression modeling. As previously said, the goal was to reduce the computational load related to the Gram matrix.

We assume that the data consists of n observations on a response variable y (which are collected in a n-length vector) and a set of p covariates which, for the same n units, are collected in the $n \times p$ matrix $\mathbf{X}$. $\mathbf{X}$ is assumed to be of full rank. The model $\mathbf{y} = \mathbf{X}\beta + \varepsilon$ is assumed to hold for the data and the goal is to find the least squares estimate for β i.e. the vector $\mathbf{b}$ that minimizes the following function:

$$\min_{\mathbf{b}} \| \mathbf{Xb} - \mathbf{y} \|^2$$

It is well known that the solution to this problem is given by

$$\mathbf{b} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

The solution only depends on the Gram matrix $\mathbf{X}^\top\mathbf{X}$ and on the marginal association $\mathbf{X}^\top\mathbf{y}$.

In order to review the theory related to the use of sketching methods in regression, the following quantities are worth introducing.

Let $TSS = \mathbf{y}^\top\mathbf{y}$; $RSS = \| \mathbf{Xb} - \mathbf{y} \|_2^2$; $MSS = \| \mathbf{Xb} \|_2^2$ and $R^2 = MSS/TSS$

where TSS, RSS and MSS are the total, residual and model sum of squares respectively and R^2 is the multiple linear correlation coefficient.

Sarlós (2006) and Woodruff (2014), using the concept of ε-subspace embedding, proved that the linear regression problem can be rephrased in terms of sketched matrices:

$$\min_{\mathbf{b}} \| SX\mathbf{b} - S\mathbf{y} \|^2 \Rightarrow \min_{\mathbf{b}} \| \widetilde{X}\mathbf{b} - \widetilde{\mathbf{y}} \|^2$$

and the solution is:

$$\mathbf{b_s} = (\widetilde{X}^\top \widetilde{X})^{-1} \widetilde{X}^\top \widetilde{\mathbf{y}}$$

where S is the Sketching matrix and $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{y}}$ are the sketched covariate matrix and response vector respectively.

$\mathbf{b_s}$ is the so called **complete sketching** estimator for the vector of linear regression coefficients.

Many theoretical results aimed at studying the properties of the sketched vector of regression coefficients provided worst case bounds.

As a consequence of the ε-subspace embedding properties, Sarlós (2006) proved that:

$$\| \mathbf{b_s} - \mathbf{b} \|_2^2 \leq \frac{\varepsilon^2}{\sigma^2_{min}(\mathbf{X})} RSS \tag{2.1}$$

where $\sigma^2_{min}(\mathbf{X})$ is the smallest singular value of $\mathbf{X}$ and RSS is the residual sum of squares of the unsketched model.

The properties of the sketched vector of regression coefficients have recently been studied according to a statistical perspective.

Ahfock *et al.* (2017) present interesting theoretical results related to the goodness of the approximation of the sketched vector with respect to the one referred to the unsketched dataset, while Dobriban & Liu (2018) mainly address the issue of the quality of $\mathbf{b_s}$ with respect to the unknown vector of the model coefficients β.

Assuming that y and $\mathbf{X}$ are fixed, Ahfock *et al.* studied the distribution of $\mathbf{b_s}$ induced by the randomness in the sketching matrix $\mathbf{S}$ for Gaussian, Hadamard and Clarkson Woodroff sketching.

When Gaussian sketching is used it can be easily proved that, conditioning on the observed data set, the sketched data set has a multivariate normal distribution. In fact, each sketched observation is obtained as a linear combination of Guassian random variables.

Starting form this simple result, Ahfock *et al.* (2017) proved the following theorem:

Theorem 2. *Suppose* $\mathbf{b_s}$ *is computed using a Gaussian sketch and* $k > p+1$. *The conditional distribution of* $\mathbf{b_s}$ *is:*

$$\mathbf{b_s}|\tilde{\mathbf{X}},\mathbf{X},\mathbf{y} \sim N(\mathbf{b},\frac{RSS}{k}(\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}})^{-1})$$

The marginal distribution of $\mathbf{b_s}$ *is:*

$$\mathbf{b_s}|\mathbf{X},\mathbf{y} \sim Student\left(\mathbf{b},\frac{RSS}{k-(p+1)}(\mathbf{X}^\top\mathbf{X})^{-1},k-(p+1)\right).$$

This means that the structured vector of regression coefficients $\mathbf{b_s}$ is an unbiased estimator for the vector of regression coefficients $\mathbf{b}$, obtained on the original dataset.

The theorem also allows to derive exact confidence intervals for the elements of $\mathbf{b_s}$.

Things are not so straightforward for the Hadamard and Clarkson - Woodruff sketches, as they are discrete distributions over an enormous combinatorial space. The explicit finite sample distribution of the sketched estimators can be written as a sum over all these possible combinations, but such a representation is not very informative. Instead, it is useful to study the large n distribution of the estimator $\mathbf{b_s}$ to obtain an interpretable expression. An important result in Ahfock *et al.* (2017) is a conditional central limit theorem for the sketched dataset that connects the Hadamard and Clarkson-Woodruff projections to the Gaussian sketch.

Besides analyzing the quality of the regression coefficients estimated on the sketched data set with respect to the regression coefficients obtained on the full unsketched data set, Ahfock *et al.* also address strict inferential issues. While in what has been described so far the source dataset is assumed as fixed and error is introduced by the random projection only, in the classical inferential setting a source of variability at the population level is added to the projection one. The properties of the sketched regression coefficients need therefore to be studied considering expectations both with respect to the random sketching matrices $\mathbf{S}$ and with respect to sampling.

In this new setting Ahfock *et al.* proved that the vector of sketched regression coefficients $\mathbf{b_s}$ is an unbiased estimator for the population one

β and its unconditional variance after Gaussian sketching is: [1]

$$\left(\frac{n-p}{k-p-1}+1\right)\sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$$

In the same inferential framework Dobriban & Liu (2018) compared the statistical efficiency of the least squares estimator **b** and of the sketched estimator $\mathbf{b_s}$ for the model parameter β. In particular they considered variance efficiency:

$$VE(\mathbf{b_s},\mathbf{b}) = \frac{E[||\mathbf{b_s}-\beta||^2]}{E[||\mathbf{b}-\beta||^2]} \tag{2.2}$$

and prediction efficiency:

$$PE(\mathbf{b_s},\mathbf{b}) = \frac{E[||\mathbf{X}\mathbf{b_s}-\mathbf{X}\beta||^2]}{E[||\mathbf{X}\mathbf{b}-\mathbf{X}\beta||^2]}$$

which can be both regarded as a usual ratio of mean squared error of two estimators.
For VE the parameter is β while for PE it is the regression function $\mathbf{X}\beta$. Both measures of efficiency are greater than or equal to 1, and smaller is better.
Finally they studied out-of-sample efficiency defined as:

$$OE(\mathbf{b_s},\mathbf{b}) = \frac{E[(\mathbf{x}_t^\top\mathbf{b_s}-y_t)^2]}{E[(\mathbf{x}^\top\mathbf{b}-y_t)^2]}$$

They considered an asymptotic framework in which both the number of variables p and the sample size n tend to infinity, and their aspect ratio converges to a constant. The size k of the sketched data is also proportional to n. Under these asymptotics they found the limits of the relative efficiencies under various conditions on **X** and **S**.
In particular, relying on results from asymptotic random matrix theory and free probability theory, they found very simple expressions for the

[1] The variance is obtained using the law of the total variance. Our result is slightly different from Ahfock *et al.*'s one as they assume $var_\varepsilon\{E_\mathbf{s}(\mathbf{b_s}|\mathbf{y},\mathbf{X})\}=0$ while we think it is equal to $\sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$ as $E_\mathbf{s}(\mathbf{b_s}|\mathbf{y},\mathbf{X})$ is **b**, i.e the original least squares estimator.
Also the degrees of freedom in Ahfock *et al.* contain a typo and should be $k-p-1$.

relative efficiencies in the context of iid sketching (of which the Guassian sketching is a special case) and of Haar/Hadamard (i.e. orthogonal sketching). They also considered uniform sampling and leverage based sampling, but we will report here on the sketching methods only.
Both for Gaussian sketching and Haar/Hadamard sketching the limits of the variance efficiency and of the prediction efficiency coincide and are respectively:

$$1+\frac{n-p}{k-p-1} \quad \text{for Gaussian sketch }^{2}$$

$$\frac{n-p}{k-p} \quad \text{for Haar/Hadamard sketch}$$

It can be easily noticed that the estimation error increases by the factor due to sketching and that it increases for Haar and Hadamard sketch less than for the iid sketch.
This is partially coherent with our finding in the previous section (which however involves the Gram matrix only and does not consider the effect of sketching on the association between the response and the covariates). Gaussian projections distort the geometry of Euclidean space due to their non-orthogonality and this in turn degrades the performance of OLS even if we do not reduce the sample size.
The difference between the efficiency of the different sketching methods decreases as the size of the sketched data sets decreases.
The limits for OE are $\frac{nk-p^2}{n(k-p)}$ and $\frac{k(n-p)}{n(k-p)}$ respectively for the Gaussian and the orthogonal sketching.

In the same paper, in which they studied how the regression coefficients obtained after sketching both **X** and **y** approximate the ones referred to the unsketched dataset, Ahfock *et al.* (2017) also studied what they called **partial sketching**, which has been introduced in the literature by Dhillon *et al.* (2013) and Pilanci & Wainwright (2016).
As the computational burden mainly affects the Gram Matrix, one can simply sketch the design matrix **X** while leaving the response variable **y** unchanged. The least squares problem can be rephrased as

$$\min_{\mathbf{b}} \parallel \mathbf{SXb}-\mathbf{y} \parallel^2 \Rightarrow \min_{\mathbf{b}} \parallel \widetilde{\mathbf{X}}\mathbf{b}-\mathbf{y} \parallel^2$$

[2] This result is coherent with our result in note 1.

and the solution is the so-called vector of partially sketched regression coefficients:

$$\mathbf{b_p} = (\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})^{-1} \mathbf{X}^\top \mathbf{y}$$

Ahfock *et al.* proved that:

$$\| \mathbf{b_p} - \mathbf{b} \|^2 \leq \frac{4\,\varepsilon^2}{\sigma^2_{min}(\mathbf{X})}\, MSS \tag{2.3}$$

i.e. the goodness of the approximation of the partially sketched regression coefficients with respect to the ones obtained on the full data set is upper bounded by a quantity that depends on the unsketched model sum of squares *MSS*. This means that when the model sum of squares is low partial sketching provides a better approximation than complete sketching; on the contrary when the error sum of squares is low complete sketching should be preferred.
When dealing with Gaussian sketching, the following results hold:

$$\widetilde{\mathbf{X}} \,|\, \mathbf{X} \sim MN(\mathbf{0}_{k\times p}, \mathbf{I}_k, \frac{1}{k}\mathbf{X}^\top\mathbf{X}) \tag{2.4}$$

i.e. $\tilde{\mathbf{X}}$ is a matrix variate normal random variable. Therefore:

$$\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}} \,|\, \mathbf{X} \sim Wishart(k, \mathbf{X}^\top\mathbf{X}/k) \tag{2.5}$$

$$\text{and} \quad (\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})^{-1} \,|\, \mathbf{X} \sim InvWishart(k, k(\mathbf{X}^\top\mathbf{X})^{-1}) \tag{2.6}$$

Useful properties of the Wishart and the inverse-Wishart random variables are reported in Appendix A. We will refer to them in the following.
$\mathbf{b_p} = (\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})^{\mathbf{-1}} \mathbf{X}^\top \mathbf{y}$ is therefore a linear combination of the elements of an inverse Wishart random variable.
Its distribution is unknown but it is possible to compute its expected value and its variance.
According to property $A6(i)$ in the appendix:

$$E\left[(\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})^{-1}\right] = \frac{1}{k-p-1}(\mathbf{X}^\top\mathbf{X})^{-1}.$$

From this:

$$E_s\left[\mathbf{b_p} \,|\, \mathbf{y}, \mathbf{X}\right] = E\left[(\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})^{-1}\mathbf{X}^\top\mathbf{y} \,|\, \mathbf{y}, \mathbf{X}\right] = \frac{k}{(k-p-1)}\mathbf{b} \tag{2.7}$$

easily derives. The subscript s stresses the fact that the expected value is computed with respect to all possible sketching matrices.
This result means that the partially sketched vector of regression coefficients $\mathbf{b_p}$ is a biased estimator of the corresponding vector $\mathbf{b}$ referred to the full data set.
The unbiased estimator can therefore be obtained as:

$$\mathbf{b_p^*} = \frac{(k-p-1)}{k}(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-\mathbf{1}}\mathbf{X}^\top\mathbf{y}.$$

Ahfock *et al.* also derived the variance both for the biased and the unbiased vector of partially sketched regression coefficients:

$$V(\mathbf{b_p}) = \frac{k^2}{(k-p)(k-p-1)(k-p-3)}(MSS(\mathbf{X}^\top\mathbf{X})^{-1} + \frac{k-p+1}{k-p-1}\mathbf{b}\mathbf{b}^\top) \tag{2.8}$$

$$V(\mathbf{b_p^*}) = \frac{(k-p-1)}{(k-p)(k-p-3)}(MSS(\mathbf{X}^\top\mathbf{X})^{-1} + \frac{k-p+1}{k-p-1}\mathbf{b}\mathbf{b}^\top) \tag{2.9}$$

Their proof is a little bit laborious. We have obtained the same result in a simpler way by relying on the result in Haff (1979) which is reported in $A6(iii)$ in the appendix.
As:

$$V(\mathbf{b_p}) = E(\mathbf{b_p}^2) - E(\mathbf{b_p})^2$$

and, according to (2.7):

$$E(\mathbf{b_p})^2 = \frac{k^2}{(k-p-1)^2}\mathbf{b}\mathbf{b}^\top$$

we only need to find a suitable expression for $E(\mathbf{b_p}^2)$ in order to obtain the result.

$$E(\mathbf{b_p}^2) = E(\mathbf{b_p}\mathbf{b_p}^\top) = E\left[(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-\mathbf{1}}\mathbf{X}^\top\mathbf{y}\mathbf{y}^\top\mathbf{X}(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-\mathbf{1}}\right]$$

which, after setting $\mathbf{X}^\top\mathbf{y}\mathbf{y}^\top\mathbf{X} = \mathbf{A}$ and using the property in $A6(iii)$, becomes:

$$E\left[(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}\mathbf{A}(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}\right] = \frac{1}{(k-p)(k-p-1)(k-p-3)}tr(k(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{A})k(\mathbf{X}^\top\mathbf{X})^{-1} + \\ + \frac{1}{(k-p)(k-p-3)}k(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{A}(\mathbf{X}^\top\mathbf{X})^{-1}k$$

But:

$$(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{A}(\mathbf{X}^\top\mathbf{X})^{-1} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}\mathbf{y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1} = \mathbf{b}\mathbf{b}^\top$$

Moreover:

$$\begin{aligned} tr[k(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{A})](\mathbf{X}^\top\mathbf{X})^{-1} &= k^2 tr[(\mathbf{X}^\top\mathbf{X})^{-1}X^\top\mathbf{y}\mathbf{y}^\top\mathbf{X}](\mathbf{X}^\top\mathbf{X})^{-1} \\ &= k^2 tr[\mathbf{b}\mathbf{y}^\top\mathbf{X}](\mathbf{X}^\top\mathbf{X})^{-1} \end{aligned}$$

and, after applying trace properties and recognizing in $\mathbf{b}^\top\mathbf{X}^\top\mathbf{y}$ the model sum of squares for the full data set, it becomes:

$$k^2 tr[\mathbf{b}^\top\mathbf{X}^\top\mathbf{y}](\mathbf{X}^\top\mathbf{X})^{-1} = k^2 MSS(\mathbf{X}^\top\mathbf{X})^{-1}$$

So:

$$E(\mathbf{b_p}^2) = \frac{k^2}{(k-p)(k-p-1)(k-p-3)} MSS(\mathbf{X}^\top\mathbf{X})^{-1} + \frac{k^2}{(k-p)(k-p-3)}\mathbf{b}\mathbf{b}^\top$$

The variance then becomes:

$$\begin{aligned} V(\mathbf{b_p}) &= \frac{k^2}{(k-p)(k-p-1)(k-p-3)} MSS(\mathbf{X}^\top\mathbf{X})^{-1} + \\ &+ \frac{k^2}{(k-p)(k-p-3)}\mathbf{b}\mathbf{b}^\top - \frac{k^2}{(k-p-1)^2}\mathbf{b}\mathbf{b}^\top \\ &= \frac{k^2}{(k-p)(k-p-1)(k-p-3)} (MSS(\mathbf{X}^\top\mathbf{X})^{-1} + \frac{k-p+1}{k-p-1}\mathbf{b}\mathbf{b}^\top) \end{aligned}$$

The variance of $\mathbf{b_p^*}$ can be derived accordingly.
It is immediately evident that the variance of the partially sketched regression coefficients depends on the model sum of squares while the one of the completely sketched regression coefficients depends on the residual sum of squares. This means that when R^2 is close to 1 complete sketching can be more efficient than partial sketching while when R^2 is close to zero the latter should be preferred.
In order to deal with intermediate situations, Ahfock *et al.* (2017) propose a combined estimator which relies on the incorrelation between $\mathbf{b_p^*}$ and $\mathbf{b_s}$.
As previously said, the explicit form of the sampling distribution of the partially sketched regression coefficients is hard to obtain. However, by

making a connection with method of moments estimation, Ahfock *et al.* also established asymptotic normality of both $\mathbf{b_p}$ and $\mathbf{b_p^*}$ as k tends to infinity. This motivates the construction of approximate confidence intervals.

They also proved that, asymptotically, the Hadamard and Clarkson-Woodruff sketches should have similar mean and variance properties to the Gaussian partially sketched estimator.

Dobriban & Liu (2018) did not address the issue related to the efficiency of partial sketching in the unconditional setting, but Ahfock *et al.* proved that $\mathbf{b_p^*}$ is also unbiased for the population parameter β, and its variance is:

$$\begin{aligned} V_\varepsilon(\mathbf{b_p^*}) = \frac{(k-p-1)}{(k-p)(k-p-3)}\{(p\sigma^2+n\tau^2)(\mathbf{X}^\top\mathbf{X})^{-1}+ \\ +(\frac{k-p+1}{k-p-1}\sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}+\frac{k-p+1}{k-p-1}\beta\beta^\top)\} \end{aligned}$$

where σ^2 is the variance of the error term and $\tau^2 = ||\mathbf{X}\beta||_2^2/n$ represents the average mean function sum of squares.

When compared with the unconditional variance of $\mathbf{b_s}$, this variance tells us that again $\mathbf{b_s}$ is more efficient when the signal to noise ratio is high and $\mathbf{b_p^*}$ is more efficient when the signal to noise ratio is low.

This is confirmed also when variance efficiency of $\mathbf{b_p^*}$ in estimating β in (2.2) is measured with respect to the ordinary least squares estimator.

Chapter 3

Matrix Sketching for LDA

3.1 Introduction

The main contribution of this thesis is the proposal of applying matrix sketching in linear discriminant analysis. As illustrated in chapter 1 LDA can be seen as a particular regression problem, therefore the same computational load inherent in the computation of the Gram matrix involved in multiple linear regression also affects LDA.

As the response variable for two group LDA is a binary variable, partial sketching represents the only viable solution. Sketching the response variable too (i.e. linearly combining the response values) would lead to loose the information on group membership, thus preventing any possible classification.

According to the partial sketching approach, only the Gram matrix is sketched while the relationship between the predictor variables and the response is left unchanged.

By applying matrix sketching to the Gram matrix involved in (1.9), (1.10), (1.11) and (1.12), the following expression for the partially sketched linear discriminant direction can be obtained:

$$\mathbf{b_p} = (\widetilde{\mathbf{X}}^{\top}\widetilde{\mathbf{X}})^{-\mathbf{1}}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)\frac{n_0 n_1}{n_0 + n_1} \tag{3.1}$$

$$\mathbf{a_p} = \gamma_p \mathbf{b_p} \tag{3.2}$$

where:

$$\gamma_p = (n-2)\,\frac{(n_0+n_1)}{n_0 n_1} + (\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)^\top W_{sk}^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0) \tag{3.3}$$

$$= (n-2)\,\frac{n}{n_0 n_1}\,\frac{1}{1-\frac{n_0 n_1}{n}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)^\top(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-\mathbf{1}}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)} \tag{3.4}$$

Equivalently $\mathbf{a_p}$ can be written as:

$$\mathbf{a_p} = W_{sk}^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0) = (n-2)(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}}-B)^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0) = \gamma_p\,\mathbf{b_p} \tag{3.5}$$

3.2 Theoretical Results

In analogy with the approach followed in the context of multiple regression, we have derived an upper bound for the squared Euclidean distance between the vector of the partially sketched linear discriminant coefficients $\mathbf{a_p}$ and the linear discriminant direction $\mathbf{a}$ referred to the unsketched dataset:

Theorem 3. *Suppose that $\widetilde{\mathbf{X}}$ is an $\varepsilon-$subspace embedding of $\mathbf{X}$, with $0<\varepsilon<0.5$. Then the following bound holds:*

$$\|\mathbf{a}-\mathbf{a_p}\|_2^2 \le d_M^2\left[(n-2)+\frac{n_0 n_1}{n}d_M^2\right]\frac{1}{\sigma_{min}^2(\mathbf{X})}4\varepsilon^2 \tag{3.6}$$

Proof.

$$\|\mathbf{a}-\mathbf{a_p}\|_2 = \|\gamma\mathbf{b}-\gamma_p\mathbf{b_p}\|_2$$
$$= \left|\frac{n}{n_0 n_1}(n-2)\right|\,\|\frac{1}{1-\frac{n_0 n_1}{n}\delta^2}\mathbf{b}-\frac{1}{1-\frac{n_0 n_1}{n}\delta_p^2}\mathbf{b_p}\|_2^2$$

where: $\delta_p^2 = (\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)^\top(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-\mathbf{1}}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\in\mathbb{R}$

and $\delta^2 = (\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)^\top(\mathbf{X}^\top\mathbf{X})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)$

Let $\mathbf{X}=\mathbf{U}\mathbf{D}\mathbf{V}^\top$ be the singular value decomposition of $\mathbf{X}$. Then:

$$\mathbf{b} = (\mathbf{X}^\top\mathbf{X})^{-\mathbf{1}}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\frac{n_0 n_1}{n}$$
$$= ((\mathbf{U}\mathbf{D}\mathbf{V}^\top)^\top\mathbf{U}\mathbf{D}\mathbf{V}^\top)^{-\mathbf{1}}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\frac{n_0 n_1}{n}$$
$$= \mathbf{V}\mathbf{D}^{-\mathbf{1}}\mathbf{D}^{-\mathbf{1}}\mathbf{V}^\top(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\frac{n_0 n_1}{n}, \quad \text{because } \mathbf{U}^\top\mathbf{U}=I_p$$

Similarly, $\mathbf{b_p}$ can be written as:

$$\begin{aligned}\mathbf{b_p} &= (\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-\mathbf{1}}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\,\frac{n_0\,n_1}{n}\\ &= ((\mathbf{SX})^\top\mathbf{SX})^{-\mathbf{1}}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\,\frac{n_0\,n_1}{n}\\ &= \mathbf{VD^{-1}}(\mathbf{U}^\top\mathbf{S}^\top\mathbf{SU})^{-\mathbf{1}}\mathbf{D^{-1}V}^\top(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\,\frac{n_0\,n_1}{n}\end{aligned}$$

So:

$$\begin{aligned}\|\mathbf{a}-\mathbf{a_p}\|_2 = (n-2)\,\|&\frac{1}{1-\frac{n_0n_1}{n}\,\delta^2}\mathbf{VD^{-1}D^{-1}V}^\top(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)+\\ &-\frac{1}{1-\frac{n_0n_1}{n}\,\delta_p^2}\mathbf{VD^{-1}}(\mathbf{U}^\top\mathbf{S}^\top\mathbf{SU})^{-\mathbf{1}}\mathbf{D^{-1}V}^\top(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\|_2\\ &= (n-2)\|\mathbf{VD^{-1}}\left[\frac{1}{1-\frac{n_0n_1}{n}\,\delta^2}\,\mathbf{I}-\frac{1}{1-\frac{n_0n_1}{n}\,\delta_p^2}(\mathbf{U}^\top\mathbf{S}^\top\mathbf{SU})^{-\mathbf{1}}\right]\mathbf{D^{-1}V}^\top\\ &(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\|_2\end{aligned}$$

Let:

$$\xi = \frac{1}{1-\frac{n_0n_1}{n}\,\delta^2}\in\mathbb{R},\ \text{and}\ \psi = \frac{1}{1-\frac{n_0n_1}{n}\,\delta_p^2}\in\mathbb{R},\ \text{with both}\ \psi,\xi>0$$

Then:

$$\begin{aligned}\|\mathbf{a}-\mathbf{a_p}\|_2 &= (n-2)\|\mathbf{VD^{-1}}\left[\xi\,\mathbf{I}-\psi(\mathbf{U}^\top\mathbf{S}^\top\mathbf{SU})^{-\mathbf{1}}\right]\mathbf{D^{-1}V}^\top(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\|_2\\ &\le (n-2)\|\mathbf{VD}^{-1}\|_2\|\xi\,\mathbf{I}-\psi(\mathbf{U}^\top\mathbf{S}^\top\mathbf{SU})^{-\mathbf{1}}\|_2\,\|\mathbf{D^{-1}V}^\top(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\|_2\end{aligned}$$

As: $\|\mathbf{VD}^{-1}\|_2 = \frac{1}{\sigma_{min}(\mathbf{X})}$, where $\sigma_{min}(\mathbf{X})$ is the minimum singular value of $\mathbf{X}$,

and: $$\begin{aligned}\|\mathbf{D^{-1}V}^\top(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\|_2^2 &= (\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)^\top\mathbf{VD^{-1}D^{-1}V}^\top(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\\ &= (\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)^\top(\mathbf{X}^\top\mathbf{X})^{-\mathbf{1}}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0) = \delta^2\end{aligned}$$

$\Rightarrow$ $\|\mathbf{D^{-1}V}^\top(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\|_2 = \delta$ then:

$$\|\mathbf{a}-\mathbf{a_p}\|_2 \le (n-2)\,\frac{1}{\sigma_{min}(\mathbf{X})}\delta\,\|\psi(\mathbf{U}^\top\mathbf{S}^\top\mathbf{SU})^{-\mathbf{1}}-\xi\,\mathbf{I}\|_2$$

We now need to upper bound the maximum singular value of the matrix $\psi(\mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\mathbf{U})^{-\mathbf{1}} - \xi\ \mathbf{I}$.
We can write:

$$\|\psi(\mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\mathbf{U})^{-\mathbf{1}} - \xi\ \mathbf{I}\|_2 = \xi\ \|\frac{\psi}{\xi}\mathbf{M}^{-\mathbf{1}} - \mathbf{I}\|_2,$$

where $\mathbf{M} = \mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\mathbf{U}$.
Since:

$$\frac{\psi}{\xi} = \frac{1 - \frac{n_0 n_1}{n}\delta}{1 - \frac{n_0 n_1}{n}\delta_p} \sim 1$$

then:

$$\|\psi(\mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\mathbf{U})^{-\mathbf{1}} - \xi\ \mathbf{I}\|_2 = \xi\ \|\mathbf{M}^{-\mathbf{1}} - \mathbf{I}\|_2,$$

The maximum absolute value of the singular values of the matrix $\mathbf{M}^{-\mathbf{1}} - \mathbf{I}$ is equal to:

$$max\left(\left|\frac{1}{\sigma_{min}(\mathbf{M})} - 1\right|, \left|\frac{1}{\sigma_{max}(\mathbf{M})} - 1\right|\right),$$

where $\sigma_{min}(\mathbf{M})$ and $\sigma_{max}(\mathbf{M})$ are the minimum and maximum singular value of $\mathbf{M}$ respectively.
Since $\mathbf{S}$ is an $\varepsilon-$subspace embedding for $\mathbf{X}$:

$$\sigma_{min}(\mathbf{M}) \geq 1 - \varepsilon, \sigma_{max}(\mathbf{M}) \leq 1 + \varepsilon$$

this implies that:

$$\frac{1}{\sigma_{max}(\mathbf{M})} \leq \frac{1}{\sigma_{min}(\mathbf{M})} \leq \frac{1}{1-\varepsilon}$$

So:

$$\begin{aligned}
\|\psi(\mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\mathbf{U})^{-1} - \xi\ \mathbf{I}\|_2 &= \xi\ \|\mathbf{M}^{-1} - \mathbf{I}\|_2 \\
&= \xi\, max(\left|\frac{1}{\sigma_{min}(\mathbf{M})} - 1\right|, \left|\frac{1}{\sigma_{max}(\mathbf{M})} - 1\right|) \\
&\leq \xi \left|\frac{1}{1-\varepsilon} - 1\right| \\
&\leq \xi\ 2\varepsilon \\
&= \frac{1}{1 - \frac{n_0 n_1}{n}\delta^2} 2\varepsilon
\end{aligned}$$

So:

$$\|\mathbf{a}-\mathbf{a_p}\|_2 \leq (n-2)\,\frac{1}{\sigma_{min}(\mathbf{X})}\,\delta\,\frac{1}{1-\frac{n_0 n_1}{n}\,\delta^2}\,2\,\varepsilon$$

Now, remembering that:

$$d_M^2(\bar{\mathbf{x}}_0,\bar{\mathbf{x}}_1) = (n-2)\,\frac{\delta^2}{1-\frac{n_0 n_1}{n}\,\delta^2}$$

then:

$$(n-2)\,\frac{1}{1-\frac{n_0 n_1}{n}\,\delta^2} = \frac{d_M^2(\bar{\mathbf{x}}_0,\bar{\mathbf{x}}_1)}{\delta^2}$$

So:

$$\|\mathbf{a}-\mathbf{a_p}\|_2 \leq \frac{d_M^2(\bar{\mathbf{x}}_0,\bar{\mathbf{x}}_1)}{\delta}\,\frac{1}{\sigma_{min}(\mathbf{X})}\,2\,\varepsilon$$

Squaring both sides, we have:

$$\|\mathbf{a}-\mathbf{a_p}\|_2^2 \leq \frac{d_M^4(\bar{\mathbf{x}}_0,\bar{\mathbf{x}}_1)}{\delta^2}\,\frac{1}{\sigma_{min}^2(\mathbf{X})}\,4\,\varepsilon^2$$

Since:

$$\delta^2 = (\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)^\top\,(\mathbf{X}^\top\mathbf{X})^{-1}\,(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0) = \frac{d_M^2}{(n-2)+\frac{n_0 n_1}{n}d_M^2}$$

the bound can finally be rewritten as:

$$\|\mathbf{a}-\mathbf{a_p}\|_2^2 \leq d_M^2\left[(n-2)+\frac{n_0 n_1}{n}d_M^2\right]\,\frac{1}{\sigma_{min}^2(\mathbf{X})}\,4\varepsilon^2$$

□

The bound essentially depends on the squared Mahalanobis distance (like $\mathbf{b_p}$ that depends on MSS) as the effect of n is compensated by $\sigma_{min}(\mathbf{X})$ according to the so called *interlacing theorem* which, when referred to singular values, states what follows:

Theorem 4. *For $A \in \mathbb{C}^{n\times p}$, let B denote A with one of its rows or columns deleted. Then:*

$$\sigma_{i+1}(A) \leq \sigma_i(B) \leq \sigma_i(A)$$

where σ_i is the i-th singular value in the decreasing sequence of the p non-zero singular values.

This implies that, as the number of rows increases, the minimum singular value does too (see for further details Horn & Johnson, 1991). To better understand the meaning of the bound, we have performed a small simulation study in which we have generated 100 samples composed of $n_0 = n_1 = 10000$ units from p-variate Normal distributions, with $p = 4$ and squared Mahalanobis distance equal to $5.6, 11.5, 23$ (these values are very close to the ones of the empirical data sets described in 3.3). The data have then been sketched to $k = 1000$. Figure 3.1 reports the boxplot of the euclidean distances between the partially sketched and the unsketched linear discriminant directions over the replicates. The results confirm that that the median distance and the variability increase as the Mahalanobis distance increases and the upper extreme of the boxplot is within the bound in (3.6).
This result is meaningful also from a statistical perspective. When the populations are separated, a plurality of different discriminant directions separates the two groups equally well, while less variability is allowed when the populations are increasingly overlapping.

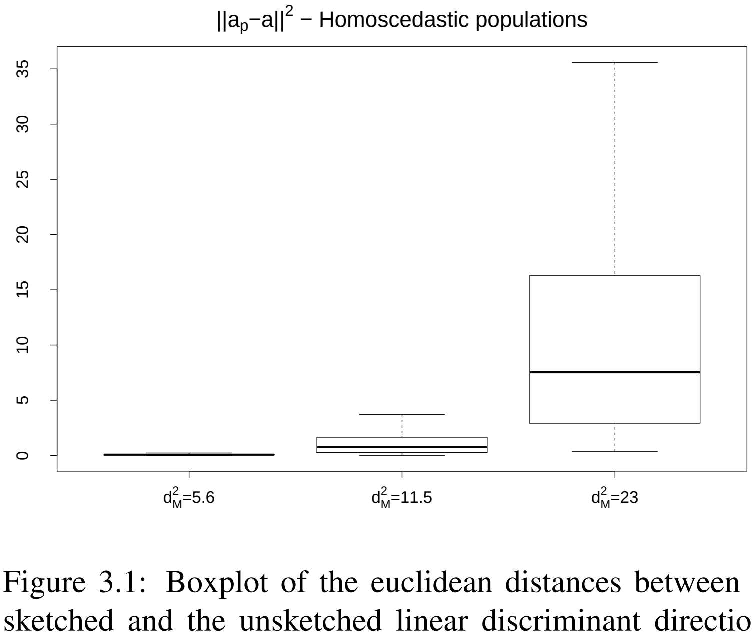

Figure 3.1: Boxplot of the euclidean distances between the partially sketched and the unsketched linear discriminant directions over 100 replicates, from multivariate normal distributions with varying Mahalanobis distance.

Besides deriving the approximation bound according to a numeric analytic perspective we have also studied the properties of the sketched discriminant direction within an inferential framework. Our goal would be first of all to derive the expected value and the variance of the sketched linear discriminant direction, in analogy to the approach followed for linear regression.

Proposition 3.1. *The random variable* $\mathbf{a_p}$ *(where randomness is due to* $\mathbf{S}$*) has finite expected value.*

Proof. According to (2.4), (2.5), (2.6):

$$\widetilde{\mathbf{X}}\,|\,\mathbf{y},\mathbf{X} \sim MN(\mathbf{0}_{k\times p},\mathbf{I}_k,\frac{1}{k}\mathbf{X}^\top\mathbf{X})$$

Therefore:

$$\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}}\,|\,\mathbf{y},\mathbf{X} \sim Wishart(k,\mathbf{X}^\top\mathbf{X}/k)$$
$$(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}\,|\,\mathbf{y},\mathbf{X} \sim InvWishart(k,k(\mathbf{X}^\top\mathbf{X})^{-1})$$

According to (3.3) and (3.4):

$$\begin{aligned}
\mathbf{a_p} &= \gamma_p\,\mathbf{b_p}\\
&= (n-2)\frac{n}{n_0 n_1}\,\frac{1}{1-\frac{n_0 n_1}{n}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)^\top(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)}\,(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\,\frac{n_0 n_1}{n}\\
&= (n-2)\,\frac{1}{1-\frac{n_0 n_1}{n}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)^\top(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)}\,(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)
\end{aligned}$$

Setting $\mathbf{u}=(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)$, $\mathbf{G}=\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}}$, $\Lambda=\mathbf{X}^\top\mathbf{X}$

$$\begin{aligned}
\mathbf{u}^\top\mathbf{a_p} &= (n-2)\,\frac{1}{1-\frac{n_0n_1}{n}\mathbf{u}^\top\mathbf{G}^{-1}\mathbf{u}}\mathbf{u}^\top\mathbf{G}^{-1}\mathbf{u}\\
&= \frac{(n-2)}{1-\frac{n_0n_1}{n}\boxed{\frac{\mathbf{u}^\top\mathbf{G}^{-1}\mathbf{u}}{\mathbf{u}^\top\Lambda^{-1}\mathbf{u}}}\mathbf{u}^\top\Lambda^{-1}\mathbf{u}}\boxed{\frac{\mathbf{u}^\top\mathbf{G}^{-1}\mathbf{u}}{\mathbf{u}^\top\Lambda^{-1}\mathbf{u}}}\mathbf{u}^\top\Lambda^{-1}\mathbf{u}
\end{aligned}$$

$$\text{where: }\boxed{\frac{\mathbf{u}^\top\mathbf{G}^{-1}\mathbf{u}}{\mathbf{u}^\top\Lambda^{-1}\mathbf{u}}} = \frac{1}{z} \sim \frac{1}{\chi^2_{n-p-1}}$$

This result is a consequence of property A4:

$$
\begin{aligned}
E\left[\mathbf{u}^\top \mathbf{a_p}\right] &= (n-2)\,E\left[\frac{1}{1-\frac{n_0 n_1}{n}\frac{1}{z}\mathbf{u}^\top\Lambda^{-1}\mathbf{u}}\ \frac{1}{z}\,\mathbf{u}^\top\Lambda^{-1}\mathbf{u}\right]\\
&= (n-2)\,E\left[\frac{1}{z-\frac{n_0 n_1}{n}\mathbf{u}^\top\Lambda^{-1}\mathbf{u}}\ \mathbf{u}^\top\Lambda^{-1}\mathbf{u}\right]\\
&= \omega\int_0^{+\infty}\frac{1}{z-\frac{n_0 n_1}{n}\underbrace{\mathbf{u}^\top\Lambda^{-1}\mathbf{u}}_{\delta}}\ \mathbf{u}^\top\Lambda^{-1}\mathbf{u}\,z^{(\frac{n-p-1}{2}-1)}e^{-\frac{z}{2}}\,dz
\end{aligned}
$$

where

$$\omega = \frac{(n-2)}{2^{\frac{n-p-1}{2}}\,\Gamma(\frac{n-p-1}{2})}$$

The singularity point $\frac{n_0 n_1}{n}\,\delta$ can be taken out as *Cauchy principal value* transformation, resulting in a finite expected value of $\mathbf{u}^\top\mathbf{a_p}$:

$$
\begin{aligned}
= \omega\lim_{\varepsilon\to 0^+}\Bigg[&\int_0^{\frac{n_0 n_1}{n}\delta-\varepsilon}\frac{1}{z-\frac{n_0 n_1}{n}\delta}\ \mathbf{u}^\top\Lambda^{-1}\mathbf{u}\,z^{(\frac{n-p-1}{2}-1)}e^{-\frac{z}{2}}\,dz+\\
&\int_{\frac{n_0 n_1}{n}\delta+\varepsilon}^{+\infty}\frac{1}{z-\frac{n_0 n_1}{n}\delta}\ \mathbf{u}^\top\Lambda^{-1}\mathbf{u}\,z^{(\frac{n-p-1}{2}-1)}e^{-\frac{z}{2}}\,dz\Bigg] < \infty
\end{aligned}
$$

□

Technical details of the proof are omitted.

This result tells us that a finite linear combination of $\mathbf{a_p}$ has finite expected value. This in turn allows to say that the variable $\mathbf{a_p}$ has finite expected value, but it doesn't give any hint on how to compute it. The distribution of $\mathbf{a}_p$ is a function of the inverse of a shifted Wishart and its moments are not known. In order to study the properties of $\mathbf{a}_p$, we derive an approximation for its expected value through its series expansion.

Proposition 3.2. *A first-order approximation of the expected value of* $\mathbf{a_p}$ *is given by:*

$$E_s(\mathbf{a_p}) = (n-2)\,E\left[(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}}-\mathbf{B})^{-1}\right](\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0) \tag{3.7}$$

$$= \left[\eta+(\alpha-\eta)\frac{(n-2)\frac{n}{n_0 n_1}}{(n-2)\frac{n}{n_0 n_1}+d_M^2(\bar{\mathbf{x}}_0,\bar{\mathbf{x}}_1)}\right]\mathbf{a} \tag{3.8}$$

where: $\alpha = \frac{k}{k-p-1}$ *and* $\eta = \frac{k^2}{(k-p)(k-p-3)}$

Proof. Recalling that Taylor's expansion of the inverse of the sum of two matrices is:

$$(\mathbf{A}+\mathbf{B})^{-1} = \mathbf{A^{-1}} - \mathbf{A}^{-1}\mathbf{BA^{-1}} + \mathbf{A}^{-1}\mathbf{BA^{-1}BA^{-1}} + \\ - \mathbf{A^{-1}BA^{-1}BA^{-1}BA^{-1}} + \cdots$$

where $\mathbf{A}$ and $\mathbf{A}+\mathbf{B}$ are invertible and $||\mathbf{A^{-1}B}|| < 1$ or $||\mathbf{BA}^{-1}|| < 1$, we derive Taylor's expansion of $(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}} - \mathbf{B})^{-1}$ and we truncate it to the first order, as higher order terms involve quadratic forms of quadratic forms of inverted Wishart random variables whose moments have not a closed form expression (for the same reason no closed form for the variance of $\mathbf{a_p}$ can be derived either).

$$E\left[(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}} - \mathbf{B})^{-1}\right] = E\left[(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}\right] + E\left[(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}\,\mathbf{B}\,(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}\right]$$

Furthermore, according to (1.13):

$$(\mathbf{X}^\top\mathbf{X} - \mathbf{B})^{-1} = (\mathbf{X}^\top\mathbf{X})^{-1} + \frac{1}{1-g}(\mathbf{X}^\top\mathbf{X})^{-1}\,\mathbf{B}\,(\mathbf{X}^\top\mathbf{X})^{-1}$$

where: $g = tr\left[(\mathbf{X}^\top\mathbf{X})^{-1}B\right] = \frac{n_0 n_1}{n}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^\top\,(\mathbf{X}^\top\mathbf{X})^{-1}\,(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0) = \frac{n_0 n_1}{n}\delta^2$
The two expressions coincide when $g = 0$.
In empirical applications of LDA g is always very close to 0, so assuming it equals 0 does not cause a too relevant loss.

Using the results in Haff (1979) reported in A6 (i) (iii) we have:

$$E\left[(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}}-\mathbf{B})^{-1}\right]=E\left[(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}\right]+E\left[(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}\,\mathbf{B}\,(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}})^{-1}\right]$$

$$=\frac{k}{(k-p-1)}(\mathbf{X}^\top\mathbf{X})^{-1}+\frac{tr(-k(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{B})}{(k-p)(k-p-1)(k-p-3)}\cdot$$

$$\cdot k(\mathbf{X}^\top\mathbf{X})^{-1}+\frac{1}{(k-p)(k-p-3)}k(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{B}(\mathbf{X}^\top\mathbf{X})^{-1}k$$

$$=\frac{k}{(k-p-1)}(\mathbf{X}^\top\mathbf{X})^{-1}-\frac{k^2g}{(k-p)(k-p-1)(k-p-3)}\cdot$$

$$\cdot(\mathbf{X}^\top\mathbf{X})^{-1}+\frac{k^2}{(k-p)(k-p-3)}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{B}(\mathbf{X}^\top\mathbf{X})^{-1}$$

$$=\frac{k}{(k-p-1)}(\mathbf{X}^\top\mathbf{X})^{-1}+\frac{k^2}{(k-p)(k-p-3)}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{B}(\mathbf{X}^\top\mathbf{X})^{-1}$$

$$=\alpha\,(\mathbf{X}^\top\mathbf{X})^{-1}+\eta\,(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{B}(\mathbf{X}^\top\mathbf{X})^{-1}$$

where we set $\alpha=\frac{k}{(k-p-1)}$ and $\eta=\frac{k^2}{(k-p)(k-p-3)}$

So, using (3.5):

$$E(\mathbf{a_p})=(n-2)E\left[(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}}-\mathbf{B})^{-1}\right](\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)$$

$$=(n-2)\left[\alpha\,(\mathbf{X}^\top\mathbf{X})^{-1}+\eta\,(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{B}(\mathbf{X}^\top\mathbf{X})^{-1}\right](\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)$$

$$=(n-2)\left[\eta\left((\mathbf{X}^\top\mathbf{X})^{-1}+(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{B}(\mathbf{X}^\top\mathbf{X})^{-1}\right)+(\alpha-\eta)(\mathbf{X}^\top\mathbf{X})^{-1}\right](\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)$$

$$=(n-2)\eta\left((\mathbf{X}^\top\mathbf{X})^{-1}+(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{B}(\mathbf{X}^\top\mathbf{X})^{-1}\right)(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)+$$

$$+(n-2)(\alpha-\eta)(\mathbf{X}^\top\mathbf{X})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)$$

$$=(n-2)\eta\,(\mathbf{X}^\top\mathbf{X}-\mathbf{B})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)+(n-2)(\alpha-\eta)(\mathbf{X}^\top\mathbf{X})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)$$

$$=\eta\,\mathbf{W}^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)+(n-2)(\alpha-\eta)(\mathbf{X}^\top\mathbf{X})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)$$

$$=\eta\,\mathbf{a}+(n-2)(\alpha-\eta)(\mathbf{X}^\top\mathbf{X})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)$$

Now, rembering that, for (1.10) and (1.11):

$$\mathbf{a}=\gamma\,\mathbf{b}=\gamma(\mathbf{X}^\top\mathbf{X})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\frac{n_0n_1}{n},\quad \text{where } \gamma=\frac{n}{n_0n_1}(n-2)+d_M^2(\bar{\mathbf{x}}_0,\bar{\mathbf{x}}_1)$$

we obtain: $(\mathbf{X}^\top\mathbf{X})^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)=\frac{1}{\gamma}\frac{n}{n_0n_1}\,\mathbf{a}=\frac{\frac{n}{n_0n_1}}{\frac{n}{n_0n_1}(n-2)+d_M^2(\bar{\mathbf{x}}_0,\bar{\mathbf{x}}_1)}\,\mathbf{a}$

So:

$$\begin{aligned}
E_s(\mathbf{a_p}) &= \eta\ \mathbf{a} + (n-2)\ (\alpha-\eta)\ \frac{\frac{n}{n_0 n_1}}{\frac{n}{n_0 n_1}(n-2) + d_M^2(\bar{\mathbf{x}}_0, \bar{\mathbf{x}}_1)}\ \mathbf{a} \\
&= \left[\eta\ + (\alpha-\eta)\ \frac{\frac{n}{n_0 n_1}(n-2)}{\frac{n}{n_0 n_1}(n-2) + d_M^2(\bar{\mathbf{x}}_0, \bar{\mathbf{x}}_1)}\right]\ \mathbf{a}
\end{aligned}$$

□

This result tells us that $\mathbf{a_p}$ is a biased estimator of $\mathbf{a}$ and that the bias depends again on the Mahalanobis distance between the two groups in the full dataset. De-biasing $\mathbf{a_p}$ would involve computing the Mahalanobis distance in the original data set, thus making the use of sketching meaningless. For this reason in the applications that follow we will not make any correction to $\mathbf{a_p}$.

It is worth noting that while $(\mathbf{X}^\top\mathbf{X} - \mathbf{B})^{-1}$ is positive definite by definition, nothing guarantees that $(\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}} - \mathbf{B})^{-1}$ is also positive definite because sketching the Gram matrix only (i.e. using partial sketching) breaks the link between the sketched total sum of squares and the unsketched between group sum of squares. In empirical applications a check should be made on the positive definiteness of $(\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}} - \mathbf{B})^{-1}$ and solutions that deviate from it should be discarded.

3.3 Real data applications

The goal of supervised classification is to definite an assignment rule that has the best accuracy. We expect that sketching can cause a loss in terms of accuracy with respect to the rule obtained on the full data set. In order to better understand how sketching impacts on the accuracy of LDA we have analyzed several real data sets with different degrees of sketching and different sketching methods.
Each data set has been randomly split in two parts: 75% of the units for both classes constituted the training set and the remaining 25% formed the test set. The procedure was repeated 200 times. The values in the table represent the median of the quantity of interest over the 200 replicates.
The datasets have markedly different sizes. We have purposely chosen

also small data set, which in principle would not require sketching, in order to have a more detailed picture of how sketching works.
Here we present the results on four of them which are briefly described in the following.

- `iris`: the famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. Here the samples are labeled as virginica vs. nonvirginica.
- `vehicle`: data include the silhouettes measured by the HIPS (Hierarchical Image Processing System) of 846 vehicles, extracted in 18 features. The aim is to distinguish the *vans* from the other vehicles. (`https://archive.ics.uci.edu/ml/datasets/Statlog+(Vehicle+Silhouettes)`).
- `banknotes`: data extracted from 1372 images of genuine and forged banknote-like specimens. Wavelet Transform tool were used to extract 4 features from images. (`http://archive.ics.uci.edu/ml`).
- `mammography`: the Mammography dataset (Woods *et al.*, 1993) has 6 attributes and 11,183 samples that are labeled as noncalcification and calcifications.

Table 3.1: `iris` dataset, n=150 - Accuracy median values (over 200 replications)

	Gauss	CW	Haar	Hadamard
Unsketched	0.92	0.92	0.92	0.92
Sketched - $k = n$	0.92	0.92	0.89	0.92
Sketched - $k = 0.75n$	0.89	0.89	0.89	0.89
Sketched - $k = 0.50n$	0.89	0.89	0.86	0.86
Sketched - $k = 0.25n$	0.86	0.86	0.82	0.84

Table 3.2: `vehicles` dataset, n=846 - Accuracy median values (over 200 replications)

	Gauss	CW	Haar	Hadamard
Unsketched	0.95	0.95	0.95	0.95
Sketched - $k = n$	0.95	0.95	0.95	0.95
Sketched - $k = 0.75n$	0.95	0.95	0.95	0.95
Sketched - $k = 0.50n$	0.94	0.94	0.93	0.93
Sketched - $k = 0.25n$	0.93	0.93	0.89	0.89

Table 3.3: `banknotes` data, n=1372 - Accuracy median values (over 200 replications)

	Gauss	CW	Haar	Hadamard
Unsketched	0.98	0.98	0.98	0.98
Sketched - $k = n$	0.98	0.97	0.95	0.97
Sketched - $k = 0.75n$	0.98	0.97	0.95	0.97
Sketched - $k = 0.50n$	0.98	0.97	0.95	0.97
Sketched - $k = 0.25n$	0.97	0.97	0.94	0.97

Table 3.4: `mammography` data, n=11,183 - Accuracy median values (over 200 replications)

	Gauss	CW	Haar	Hadamard
Unsketched	0.98	0.98	0.98	0.98
Sketched - $k = n$	0.98	0.98	0.98	0.98
Sketched - $k = 0.75n$	0.98	0.98	0.97	0.97
Sketched - $k = 0.50n$	0.98	0.98	0.97	0.96
Sketched - $k = 0.25n$	0.98	0.98	0.97	0.95

All the empirical results show that sketching has a very little impact on accuracy, which remains almost unchanged in all the cases, even when, for $k = 0.25n$, the data set size is reduced to one fourth. Gaussian and Clarkson-Woodruff sketches seem to guarantee the best performances.

The only case in which the performances are really deteriorated is for the Iris data (see Table 3.1). However table 3.5 shows that the number of available units in that case is below the limit required by Johnson and Lindenstrauss Lemma even for a precision corresponding to $\varepsilon = 0.49$ which yields the worst possible approximation. In all the other cases the accuracies are preserved even for sketching degrees that do not allow to guarantee a good approximation of the discriminant direction. The table shows the value of k required by Johnson Lindenstrauss Lemma for the different number of variables in the different datasets that have been analyzed and for different quality of approximation corresponding to different values of ε. The larger the value of ε the worse the approximation.

Table 3.5: Minimum value of $k = \dfrac{20 \log p}{\varepsilon^2}$ required to obtain a given approximation of ε for the different number of variables in the empirical data sets. In bold the values of k compatible with the observed number of units ($n_{training} = 112$ for *iris*, $n_{training} = 634$ for *vehicle*, $n_{training} = 1029$ for *banknotes*, $n_{training} = 8387$ *mammography*).

	$\varepsilon = 0.05$	0.1	0.2	0.3	0.4	0.49
`iris` & `banknotes` - $p = 4$	11090	2773	**693**	308	173	**115**
`vehicle` - $p = 18$	22123	5781	1445	642	**361**	241
`mammography` - $p = 6$	14334	**3584**	896	398	224	149

Chapter 4

Matrix Sketching for imbalanced classes

4.1 Introduction

In many practical contexts, observations have to be classified into two classes of remarkably distinct size.
In such cases, many established classifiers often trivially classify instances into the majority class achieving an optimal overall misclassification error rate.
This leads to poor performance in classifying the minority class, the correct identification of which is usually of more practical interest.
The presence of imbalanced classes in the big data context also poses relevant computational issues. If the dataset contains thousands or millions of observations from the majority class for each example from the minority one, many of the majority class observations are redundant. Their presence increases the computational cost with no advantage in terms of classification accuracy.
The problem of imbalanced classes is very common in modern classification problems and has received a great attention in the machine learning literature (Chawla *et al.*, 2004).

The error rate, or its complement, accuracy, is the most widely used measure of classifier performance. However, it inevitably favors the majority class when the misclassification error has the same importance for the two classes. On the contrary, when the error in the minority class is more important than the one of the majority class, the receiver operating characteristic (ROC) curve and the area under the curve (AUC) are commonly suggested.

The ROC curve is a plot of the true positive rate (*sensitivity*) versus the false positive rate (*1−specificity*) and hence a higher AUC generally indicates a better classifier.

The ROC is obtained by varying the discriminant threshold, while the error rate is obtained at an optimal discriminant threshold. Therefore, AUC is independent of the discriminant threshold, while the accuracy is not.

The literature on the problem of supervised classification is very broad and methodological solutions follow two main streams. People either suggest to modify the loss function used in the construction of the classification rule or propose to re-balance the data.

The first solution requires, in most of the cases, the definition of a loss function that is specific for the case at hand and therefore not easily generalizable to different empirical problems. Re-balancing strategies are more general and not problem specific. That explains their great success in applied research and the focus on explaining their performances and on improving them.

Re-balancing the class sizes in the training dataset, is usually obtained either by oversampling the minority class or by under-sampling the majority class, or by a combination of both. The rebalanced data are then used to train the classifiers.

As far as two-class linear discriminant analysis is concerned, the problem has been addressed, among others, by Xie & Qiu (2007), Xue & Titterington (2008), Xue & Hall (2014).

Through a wide simulation study supported by theoretical considerations, Xue & Titterington (2008) showed that AUC generally favors balanced data but the increase in the median AUC for LDA after rebalancing is relatively small. On the contrary, error rate favors the original data and re-balancing causes a sharp increase in the median error rate. They also stress that re-balancing affects the performances of LDA in both the equal and unequal covariance case.

Xue & Hall (2014) proved that, in the Gaussian case, using the rebalanced training data can often increase the area under the ROC curve (AUC) for the original, imbalanced test data. In particular they demonstrate that, at least for LDA, there is an intrinsic, positive relationship between the re-balancing of class sizes and the improvement of AUC and the largest improvement in AUC can be achieved, asymptotically, when the two classes are fully rebalanced to be of equal sizes.

$$AUC_{max} = \Phi\left(\sqrt{(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^\top (\mathbf{S_1} + \mathbf{S_0})^{-\mathbf{1}} (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)}\right)$$

where, now and henceforth, the subscript 1 identifies the minority class. However, when the two Gaussian classes have similar covariance matrices, re-balancing class sizes only provides a little improvement in AUC for LDA.

Moreover, re-balancing class sizes may not improve AUC when LDA is applied to non-Gaussian data.

In both the above mentioned papers re-balancing is obtained either by randomly undersampling the largest class or by randomly oversampling the smallest one.

It has however been argued that random undersampling may lose some relevant information while randomly oversampling with replacement the smallest class may lead to overfitting.

To avoid these drawbacks, solutions focusing on the border between the classes have been suggested. Mani & Zhang (2003) proposed selecting majority class examples whose average distance to its three nearest minority class examples is smallest. A similar approach is suggested by Fithian & Hastie (2014) in the context of logistic regression. They proposed a method of efficient subsampling by adjusting the class balance locally in feature space via an acceptance-rejection scheme. The proposal generalizes case-control sampling, using a pilot estimate to preferentially select examples whose responses (i.e. class membership identifiers) are conditionally rare given their features.

With special reference to classification trees and naive-Bayes classifiers Chawla *et al.* (2002) proposed a strategy that combines random undersampling of the majority class with a special kind of oversampling for the minority one. They try to improve upon literature results according to which undersampling the majority class leads to better classifier performance than oversampling and combining the two does not produce much improvement with respect to simple undersampling.

They designed an oversampling approach which creates synthetic examples (SMOTE - Synthetic minority oversampling technique) rather than oversampling with replacement. The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the *k* minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the *k* nearest neighbors are randomly chosen. The synthetic examples allow to create larger and less specific decision regions, thus overcoming the overfitting effect inherent in random oversampling. SMOTE oversampling is combined with majority class undersampling. The idea of creating synthetic examples has been followed also by Menardi & Torelli (2014) who proposed a method they called ROSE-*Random OverSampling Examples* (see, for a description of the corresponding R package, Lunardon *et al.*, 2014). In this solution, units from both classes are generated by resorting to a smoothed bootstrap approach. A unimodal density is centered on randomly selected observations and new artificial data are randomly generated from it. The key parameter of the procedure is the dispersion matrix of the chosen unimodal density which plays the role of smoothing parameter. The full dataset size is often kept fixed while allowing half of the units to be generated from the minority class and of half from the majority one. The method is applied to classification trees and logit models.

4.2 Rebalancing through Sketching

In the previous chapters we have seen that sketching preserves the scalar product while reducing the data set size. As the sketched data are obtained through random linear combinations of the original ones, most of the linear information is preserved after sketching. This means that, in the imbalanced data case, the size of the majority class can be reduced through sketching without incurring the risk of losing (too much) linear information. Sketching the majority class can therefore be considered as a theoretically sound alternative to majority class undersampling. The sketched majority class data matrix reduced to the size of the minority class will then be $\widetilde{X}_0$ $(n_1 \times p)$ and the corresponding covariance matrix will be $\mathrm{Var}(\widetilde{X}_0) = (\widetilde{X}_0^\top \widetilde{X}_0)/(n_0 - 1)$.

Sketching has been proposed as a data compression technique but, as a consequence of Johnson Lindenstrauss lemma, the scalar product preservation also holds when the sketching matrix has a number of rows that is larger than the number of original data points. This allows to think of this unconventional way of using sketching as an alternative to random

oversampling that generates synthetic new examples from the minority class (through random linear combination of all of them) while preserving the linear structure in the data. This allows to enlarge the decision area and thus to avoid overfitting. For example, in case it is desired to increase the size of the minority class so that it equals the one of the majority class the "oversketched" minority class data matrix will be $\widetilde{X}_1$ $(n_0 \times p)$ and the corresponding covariance matrix will therefore be: $\mathrm{Var}(\widetilde{X}_1) = (\widetilde{X}_1^\top \widetilde{X}_1)/(n_1 - 1)$
The rebalanced covariance matrices can then be plugged into the within group covariance matrix and used for the computation of a new linear discriminant direction.
Sketching the majority class and oversketching the minority one can also be used in a combined way.
The use of matrix sketching is undoubtedly coherent with linear discriminant analysis which is based on the Gram matrix. Its performance in combination with other classification methods is not supported by the same strong theoretical motivation and can only be assessed through empirical analysis. This will be the topic of the next section in which different sketching methods are compared.
All the different sketching methods preserve the Gram matrix even if, as shown in chapter 2, with a different goodness of approximation for different degrees of sketching. The different sketching methods can however change the data distribution. For instance Gaussian sketching tends to "gaussianize" the data and can therefore strongly distort skew data distributions. Moreover, as each linear combination is a function of all the units, potentially outlying observations impact on all the sketched data values and their effect is amplified. This effect is less evident for instance for Clarkson-Woodruff sketching which, being a sparse sketching method, only selects a few units for each random linear combination.

4.3 Empirical Results

The properties of sketching as a re-balancing method have been tested on many real datasets which differ in terms of imbalance degree. Here we report the results on the two most significant ones (`spine` and `mammography`) which have been classified both by linear discriminant analysis (see tables 4.1 and 4.2) and C4.5 classification tree (Quinlan, 2014) (see tables 4.3 and 4.4). The data set `mammography` has already been described in 3.3. The data set `spine` is composed of $p = 6$ biomechanical features used to classify $n = 310$ orthopedic patients into 2 classes (normal or abnormal). (`http://archive.ics.uci.edu/ml/datasets/vertebral+column`).
Gaussian, Hadamard and Clarkson-Woodruff sketching have been applied in order to reduce the size of the majority class to the one of the minority class and in order to increase the size of the minority class so that it is as large as the majority class one. They have also been jointly used so that the size of both classes is twice the minority class size. For this last case re-balancing through SMOTE is also performed. For comparison, ROSE with its default option of preserving the total size is considered too.
As in chapter 3, each data set has been randomly split in two parts: 75% of the units for both classes constituted the training set and the remaining 25% formed the test set. The procedure was repeated 200 times. The values in the table represent the median of the quantity of interest over the 200 replicates.
The performance of the classifiers has been measured in terms of accuracy, sensitivity, specificity and area under the ROC curve (AUC).
The code implementing our procedure is reported in the appendix, while ROSE and SMOTE have been applied using the corresponding R packages `ROSE` and `DMwR`.

Table 4.1: `spine` dataset, n=310, π_1=32% - Median values (over 200 replications)

	Accuracy	Sensitivity	Specificity	AUC
LDA	0.831	0.904	0.680	0.897
Under-Sampling	0.792	0.731	0.920	0.907
Gauss Partial Sk	0.792	0.731	0.920	0.900
CW Partial Sk	0.792	0.750	0.880	0.899
Hada Partial Sk	0.792	0.731	0.920	**0.910**
Over-Sampling	0.792	0.750	0.920	0.900
Gauss Partial OverSk	0.792	0.731	0.920	0.903
CW Partial OverSk	0.792	0.731	0.920	0.902
Hada Partial OverSk	0.792	0.731	0.920	**0.910**
ROSE	0.792	0.712	0.960	0.905
SMOTE	0.792	0.750	0.920	0.905
UndOver-Sampling Bal	0.792	0.750	0.900	0.902
Gauss Bal Sk	0.792	0.750	0.880	0.896
CW Bal Sk	0.792	0.750	0.920	0.902
Hada Bal Sk	0.792	0.731	0.920	**0.910**

Table 4.2: `mammography`, n=11,183, π_1=2.32% - Median values (over 200 replications)

	Accuracy	Sensitivity	Specificity	AUC
LDA	0.977	0.986	0.554	0.903
Under-Sampling	0.830	0.829	0.892	0.928
Gauss Partial Sk	0.827	0.826	0.907	**0.930**
CW Partial Sk	0.827	0.825	0.892	0.923
Hada Partial Sk	0.742	0.739	0.892	0.914
Over-Sampling	0.829	0.828	0.892	0.931
Gauss Partial Over Sk	0.828	0.826	0.892	**0.932**
CW Partial Over Sk	0.828	0.826	0.900	0.931
Hada Partial Over Sk	0.743	0.739	0.892	0.914
ROSE	0.977	1.000	0.000	0.878
SMOTE	0.841	0.840	0.892	0.930
UndOver-Sampling Bal	0.837	0.836	0.892	0.928
Gauss Bal Sk	0.829	0.827	0.900	**0.932**
CW Bal Sk	0.957	0.964	0.677	0.900
Hada Bal Sk	0.744	0.740	0.892	0.914

Table 4.1 and 4.2 show that, coherently with the findings in Xue and Titterington and Xue and Hall, when combined with LDA, rebalancing causes a strong decrease in the accuracy which is combined with a little increase in the AUC. However a strong increase in specificity, i.e. in the ability to correctly identify the minority class, is worth of note. In this context sketching based methods always outperform the other rebalancing methods. It does not seem to be any evidence of a systematic predominance of over, under or balanced sketching strategies.

As already said, sketching preserves the linear structure which is the core element of LDA. The good performances of sketching in this context are therefore coherent with its theoretical properties. However, when sketching methods are combined with classification methods that do not rely on the linear structure in the data, results are not so clear-cut and they seem to be strongly related to specific characteristics of the data.

Tables 4.3 and 4.4 report the results of C4.5 classification trees. While for the `spine` dataset the sketching methods perform well, for the `mammography` dataset sketching methods are strongly outperformed by standard random oversampling or undersampling methods and by ROSE.

Table 4.3: `spine` dataset, n=310, π_1=32% - Median values (over 200 replications)

	Accuracy	Sensitivity	Specificity	AUC
C4.5 Tree	0.818	0.904	0.680	0.773
Under-Sampling	0.805	0.788	0.840	0.822
Gauss Partial Sk	0.792	0.731	0.920	**0.825**
CW Partial Sk	0.792	0.750	0.880	0.817
Hada Partial Sk	0.805	0.750	0.920	**0.825**
Over-Sampling	0.818	0.846	0.720	0.795
Gauss Partial OverSk	0.805	0.788	0.840	0.813
CW Partial OverSk	0.805	0.788	0.840	0.815
Hada Partial OverSk	0.805	0.769	0.880	**0.825**
ROSE	0.792	0.712	0.960	**0.835**
SMOTE	0.805	0.808	0.800	0.805
UndOver-Sampling Bal	0.812	0.846	0.760	0.794
Gauss Bal Sk	0.805	0.750	0.920	**0.835**
CW Bal Sk	0.805	0.788	0.880	0.824
Hada Bal Sk	0.818	0.779	0.880	0.834

Table 4.4: `mammography`, n=11,183, π_1=2.32% - Median values (over 200 replications)

	Accuracy	Sensitivity	Specificity	AUC
C4.5 Tree	0.985	0.997	0.508	0.752
Under-Sampling	0.893	0.894	0.846	**0.879**
Gauss Partial Sk	0.763	0.241	0.938	0.592
CW Partial Sk	0.838	0.165	0.969	0.562
Hada Partial Sk	0.807	0.806	0.846	0.827
Over-Sampling	0.979	0.987	0.646	**0.815**
Gauss Partial OverSk	0.964	0.973	0.477	0.729
CW Partial OverSk	0.977	0.988	0.538	0.762
Hada Partial OverSk	0.931	0.937	0.692	0.813
ROSE	0.901	0.903	0.800	0.850
SMOTE	0.914	0.916	0.846	**0.879**
UndOver-Sampling Bal	0.914	0.917	0.831	0.873
Gauss Bal Sk	0.720	0.624	0.908	0.724
CW Bal Sk	0.756	0.257	0.923	0.591
Hada Bal Sk	0.839	0.838	0.815	0.830

Chapter 5

Conclusions

We studied the performances of sketching algorithms in the supervised binary classification context. In particular the use of what is called *partial sketching* in linear discriminant analysis is evaluated according to both a numeric analytic and a statistical perspective.

The discriminant direction $\mathbf{a_p}$, obtained after sketching the Gram matrix, closely approximates the one computed on the original data. The goodness of the approximation depends on the Mahalanobis distance between the two populations. We also proved that the expected value of $\mathbf{a_p}$ is finite.

The performance of linear discriminant analysis on the sketched data is compared to the one obtained on the original data in a number of real datasets, with varying degree of sketching. All the empirical results show that sketching has a very little impact on accuracy, which remains almost unchanged in all the evaluated cases, even when the data set size was reduced to one fourth.

We also addressed, through sketching, the issue of imbalanced classes, which hampers most of the common classification methods. In fact, when observations have to be classified into two classes of remarkably distinct size, many established classifiers often allocate instances into the majority class, achieving an optimal overall misclassification error rate. This leads to poor performance in classifying the minority class, the correct identification of which is usually of more practical interest.

As sketching preserves the scalar product while reducing the data set size, we can say that most of the linear information is preserved after sketching. This means that, in the imbalanced data case, the size of the majority class can be reduced through sketching without incurring the risk of losing (too much) linear information. Sketching the majority class can therefore be considered as a theoretically sound alternative to majority class undersampling.

The properties of sketching as a re-balancing method have been tested on many real datasets which differ in terms of imbalance degree and compared with other competing alternatives.

When combined with LDA, rebalancing causes a strong decrease in the accuracy which is combined with a little increase in the AUC. However a strong increase in specificity, i.e. in the ability to correctly identify the minority class, is worth of note. In this context, sketching based methods always outperform the other rebalancing methods. It does not seem to be any evidence of a systematic predominance of over, under or balanced sketching strategies.

However, when sketching is combined with classification methods that do not rely on the linear structure in the data, results are not so clear-cut and they seem to be strongly related to specific characteristics of the data.

Appendix

In the following, we recall a few theorems and theoretical results that have been used in the thesis. The first ones relate to the Wishart and inverse-Wishart random variables. We refer to Gupta & Nagar (2018) and the references therein for the proof.

A 1. *Let* $X \sim N(0, \Sigma \otimes I_n)$ *and define* $S = XX^\top, n \geq p$. *Then* $S \sim W_p(n, \Sigma)$.

A 2. *Let* $S \sim W_p(n, \Sigma)$. *Then, for* $A_{q \times p}$, *with* $rank(A) = q \leq p$, $ASA^\top \sim W_q(n, A\Sigma A^\top)$.

A 3. *Let* $S \sim W_p(n, \Sigma)$. *Then* $\frac{\mathbf{a}^\top S \mathbf{a}}{\mathbf{a}^\top \Sigma \mathbf{a}} \sim \chi^2_n$ *where* $\mathbf{a}(p \times 1) \neq 0$.

A 4. *Let* $S \sim W_p(n, \Sigma)$ *and* $\mathbf{a} \in \mathbb{R}^p, \mathbf{a} \neq 0$. *Then* $\frac{\mathbf{a}^\top \Sigma^{-1} \mathbf{a}}{\mathbf{a}^\top S^{-1} \mathbf{a}} \sim \chi^2_{n-p+1}$

A 5. *Let* $S \sim W_p(n, \Sigma)$. *Then* $E(S) = n\Sigma$.

A 6. *Let* $S \sim W_p(n, \Sigma)$, *then:*

(i) $E(S^{-1}) = \frac{1}{n-p-1}\Sigma^{-1}, n-p-1 > 0$

(ii) $cov(s^{ij}, s^{kl}) = \frac{2(n-p-1)^{-1}\sigma^{ij}\sigma^{kl} + \sigma^{ik}\sigma^{jl} + \sigma^{il}\sigma^{kj}}{(n-p)(n-p-1)(n-p-3)}, n-p-3 > 0$

(iii) $E(S^{-1}AS^{-1}) = \frac{tr(\Sigma^{-1}A)\Sigma^{-1}}{(n-p)(n-p-1)(n-p-3)} + \frac{\Sigma^{-1}A\Sigma^{-1}}{(n-p)(n-p-3)}$,
where $S^{-1} = (s^{ij}), \Sigma^{-1} = (\sigma^{ij}) and A_{p \times p}$ *is a constant positive semidefinite matrix.*

A 7. *Let* $S \sim W_p(n, \Sigma)$, *then the density of* $V = S^{-1}$ *is*

$$\{2^{\frac{1}{2}np}\Gamma_p(\frac{1}{2}n)\, det(\Sigma)^{\frac{1}{2}n}\}^{-1} det(V)^{-\frac{1}{2}(n+p+1)}\, etr(-\frac{1}{2}\Sigma^{-1}V^{-1}), V > 0.$$

R codes

Matrix sketching functions

```
hada.sk<-function(n,k){
require(extraDistr)
if (zero.row(n)!=0) n <- n + zero.row(n)
D <- diag(rsign(n))
H <- julia_eval(paste0("hadamard(",n,")"),
    need_return = c("R", "Julia"))
I <- matrix(0,k,n)
if (k>n) which.unit <- sample(1:n,k,replace=T)
else which.unit <- sample(1:n,k,replace=F)
for (i in 1:k) I[i,which.unit[i]] <- 1
return(1/sqrt(n)*I%*%H%*%D)
}

# This function returns the number of zero rows
# we need in order to compute the Hadamard matrix

zero.row <- function(n) {
if (n %in% c(1,3,5,6,9,10))
return(12-n)
esponente <- ceiling(log2(c(n,n/12,n/20)))
xx <- c(1,12,20)*2^esponente
return(min(xx-n))}

cw.sk <- function(n,k){
I <- matrix(0,k,n)
indice <- sample(1:k,n,replace=T)
for (i in 1:n) I[indice[i],i] <- rsign(1)
return(I)}

haar.sk <- function(n,k){
return(t(randortho(n,type=c("orthonormal"))[,1:k]))}

gauss.sk <- function(n,k){
return(matrix(rnorm(prod(k,n),sd=1/sqrt(k)),k,n))}
```

Classification with LDA

Data have been split into training and test data. We have applied Gaussian sketching, Hadamard sketching, Haar sketching and Clarkson-Woodruff sketching to the original data matrix **X**.

```
x <- as.matrix(x)
x <- x[c(which(y==0),which(y==1)),]
y <- y[c(which(y==0),which(y==1))]
x0 <- x[y==0,]; y0 <- y[y==0];
x1 <- x[y==1,]; y1 <- y[y==1];
n0 <- sum(y==0); n1 <- sum(y==1); n <- n0+n1;

# in case k is function of n
k <- round(r*n,0) # r varies between 0 and 1
                  # usually r={0.25,0.50,0.75}

quali.te0 <- sample(1:n0,round(n0*0.25),replace=FALSE)
quali.te1 <- sample(1:n1,round(n1*0.25),replace=FALSE)

test.x0 <- x0[quali.te0,];  test.x1 <- x1[quali.te1,];
test.y0 <- y0[quali.te0];   test.y1 <- y1[quali.te1];
te.n0 <- nrow(test.x0); te.n1 <- nrow(test.x1);

test.x <- rbind(test.x0,test.x1)
test.y <- c(test.y0,test.y1)
train.x0 <- x0[-quali.te0,]; train.x1 <- x1[-quali.te1,];
train.y0 <- y0[-quali.te0];  train.y1 <- y1[-quali.te1];
tr.n0 <- nrow(train.x0); tr.n1 <- nrow(train.x1);
train.x <- rbind(train.x0,train.x1)
train.y <- c(train.y0,train.y1)

X0 <- scale(train.x0,T,F)
X1 <- scale(train.x1,T,F)
X <- scale(train.x,T,F)
mean.tr.x0 <- colMeans(train.x0)
mean.tr.x1 <- colMeans(train.x1)
tr.n <- nrow(train.x)


# Hadamard sketching  ####
hada.skx <- hada.sk(n=tr.n,k)
hada.sk.x <- hada.skx%*%
        rbind(X,matrix(0,zero.row(tr.n),ncol(X)))

B <- (mean.tr.x0-mean.tr.x1)%*%t(mean.tr.x0-mean.tr.x1)
                                  *(tr.n0*tr.n1)/tr.n
W.sk.hada <- (t(hada.sk.x)%*%hada.sk.x-B)/(tr.n-2)

a.sk.hada <- t(solve(W.sk.hada,tol=1e-150)%*%
                              (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.sk.hada)


if (a.sk.hada%*%mean.tr.x0>a.sk.hada%*%mean.tr.x1)
    cl.sk.hada <- ifelse(train.x%*%t(a.sk.hada) <
              c(log(tr.n0/tr.n1)+1/2*a.sk.hada%*%
                    (mean.tr.x0+mean.tr.x1)),1,0)
else
    cl.sk.hada <- ifelse(train.x%*%t(a.sk.hada) >
              c(log(tr.n0/tr.n1)+1/2*a.sk.hada%*%
                     (mean.tr.x0+mean.tr.x1)),1,0)


# Clarkson-Woodruff sketching  ####
clark.skx <- clark.sk(tr.n,k)
clark.sk.x <- clark.skx%*%X
W.sk.clark <- (t(clark.sk.x)%*%clark.sk.x-B)/(tr.n-2)

a.sk.clark <- t(solve(W.sk.clark,tol=1e-150)%*%
                         (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.sk.clark)

if (a.sk.clark%*%mean.tr.x0>a.sk.clark%*%mean.tr.x1)
    cl.sk.clark <- ifelse(train.x%*%t(a.sk.clark) <
                c(log(tr.n0/tr.n1)+1/2*a.sk.clark%*%
                       (mean.tr.x0+mean.tr.x1)),1,0)
else
    cl.sk.clark <- ifelse(train.x%*%t(a.sk.clark) >
                c(log(tr.n0/tr.n1)+1/2*a.sk.clark%*%
                       (mean.tr.x0+mean.tr.x1)),1,0)
```

```
# Gaussian Sketching ####
gauss.skx <- gauss.sk(tr.n,k)
gauss.sk.x <- gauss.skx%*%X
W.sk.gauss <-(t(gauss.sk.x)%*%gauss.sk.x-B)/(tr.n-2)

a.sk.gauss <- t(solve(W.sk.gauss,tol=1e-150)%*%
                         (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.sk.gauss)

if (a.sk.gauss%*%mean.tr.x0>a.sk.gauss%*%mean.tr.x1)
    cl.sk.gauss <- ifelse(train.x%*%t(a.sk.gauss) <
                c(log(tr.n0/tr.n1)+1/2*a.sk.gauss%*%
                       (mean.tr.x0+mean.tr.x1)),1,0)
else
    cl.sk.gauss <- ifelse(train.x%*%t(a.sk.gauss) >
                c(log(tr.n0/tr.n1)+1/2*a.sk.gauss%*%
                       (mean.tr.x0+mean.tr.x1)),1,0)


# Haar Sketching ####
haar.sk <- haar.sk(n=tr.n,k)
haar.x <- haar.sk%*%train.x
B <- (mean.tr.x0-mean.tr.x1)%*%
        t(mean.tr.x0-mean.tr.x1)*(tr.n0*tr.n1)/tr.n
W.haar <- (t(haar.x)%*%haar.x-B)/(tr.n-2)

a.sk.haar <- t(solve(W.haar,tol=1e-45)%*%
                            (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.sk.haar)

if (a.sk.haar%*%mean.tr.x0>a.sk.haar%*%mean.tr.x1)
    cl.sk.haar <- ifelse(train.x%*%t(a.sk.haar) >
                c(log(tr.n0/tr.n1)+1/2*a.sk.haar%*%
                      (mean.tr.x0+mean.tr.x1)),0,1)
else
    cl.sk.haar <- ifelse(train.x%*%t(a.sk.haar) >
                c(log(tr.n0/tr.n1)+1/2*a.sk.haar%*%
                      (mean.tr.x0+mean.tr.x1)),1,0)
```

```
# Test Set

test.sk.gauss <- test.x%*%t(a.sk.gauss)
test.sk.hada <- test.x%*%t(a.sk.hada)
test.sk.clark <- test.x%*%t(a.sk.clark)
test.sk.haar <- test.x%*%t(a.sk.haar)

if (a.sk.haar%*% colMeans(train.x0) >
                   a.sk.haar%*%colMeans(train.x1))
haar.test.cl <- ifelse(test.sk.haar >
                c(log(tr.n0/tr.n1)+ 1/2*a.sk.haar%*%
       (colMeans(train.x0)+ colMeans(train.x1))),0,1)
else
    haar.test.cl <- ifelse(test.sk.haar >
                c(log(tr.n0/tr.n1)+ 1/2*a.sk.haar%*%
      (colMeans(train.x0)+ colMeans(train.x1))),1,0)


if (a.sk.hada%*%colMeans(train.x0) >
                    a.sk.hada%*%colMeans(train.x1))
test.sk.hada.cl <- ifelse(test.sk.hada <
                c(log(tr.n0/tr.n1)+ 1/2*a.sk.hada%*%
      (colMeans(train.x0)+ colMeans(train.x1))),1,0)
else
    test.sk.hada.cl <- ifelse(test.sk.hada >
               c(log(tr.n0/tr.n1)+ 1/2*a.sk.hada%*%
     (colMeans(train.x0)+ colMeans(train.x1))),1,0)

if (a.sk.clark%*%colMeans(train.x0)>
                    a.sk.clark%*%colMeans(train.x1))
    test.sk.clark.cl <- ifelse(test.sk.clark <
                c(log(tr.n0/tr.n1)+1/2*a.sk.clark%*%
      (colMeans(train.x0)+ colMeans(train.x1))),1,0)
else
    test.sk.clark.cl <- ifelse(test.sk.clark >
              c(log(tr.n0/tr.n1)+ 1/2*a.sk.clark%*%
     (colMeans(train.x0)+ colMeans(train.x1))),1,0)


if (a.sk.gauss%*%colMeans(train.x0) >
                   a.sk.gauss%*%colMeans(train.x1))
    test.gwps.cl <- ifelse(test.gauss <
             c(log(tr.n0/tr.n1)+ 1/2*a.sk.gauss%*%
     (colMeans(train.x0)+ colMeans(train.x1))),1,0)
else
    test.gwps.cl <- ifelse(test.gauss >
             c(log(tr.n0/tr.n1)+ 1/2*a.sk.gauss%*%
     (colMeans(train.x0)+colMeans(train.x1))),1,0)
```

Group-wise Partial Sketching and Over-Sketching

Classification via LDA

```
library(MCMCpack)
library(pracma)
library(DMwR)
library(Rweka)
library(extraDistr)
library(JuliaCall)
julia <- julia_setup(JULIA_HOME= "/.../julia/bin/")
julia_library("Hadamard")

x <- as.matrix(x)
x <- x[c(which(y==0),which(y==1)),]
y <- y[c(which(y==0),which(y==1))]
x0 <- x[y==0,]; y0<-y[y==0];
x1 <- x[y==1,]; y1<-y[y==1];
n0 <- sum(y==0); n1<-sum(y==1); n<-n0+n1;

which.te0 <- sample(1:n0,round(n0*0.25),replace=FALSE)
which.te1 <- sample(1:n1,round(n1*0.25),replace=FALSE)

test.x0 <- x0[which.te0,];  test.x1 <- x1[which.te1,];
test.y0 <- y0[which.te0];   test.y1 <- y1[which.te1];
te.n0 <- nrow(test.x0); te.n1 <- nrow(test.x1);

test.x <- rbind(test.x0,test.x1)
test.y <- c(test.y0,test.y1)

train.x0 <- x0[-which.te0,]; train.x1 <- x1[-which.te1,];
train.y0 <- y0[-which.te0];  train.y1 <- y1[-which.te1];
```

```
tr.n0 <- nrow(train.x0); tr.n1 <- nrow(train.x1);
train.x <- rbind(train.x0,train.x1)
train.y <- c(train.y0,train.y1)

X0 <- scale(train.x0,T,F)
X1 <- scale(train.x1,T,F)
mean.tr.x0 <- colMeans(train.x0)
mean.tr.x1 <- colMeans(train.x1)
ntr <- nrow(train.x)


# Hadamard Sketching ####
hada.sk0.und <- hada.sk(n=tr.n0,k=tr.n1)
hada.sk1.und <- hada.sk(n=tr.n1,k=tr.n1)

hada.sk.x0.und <- hada.sk0.und%*%
    rbind(X0,matrix(0,zero.row(tr.n0),ncol(X0)))
hada.sk.x1.und <- hada.sk1.und%*%
    rbind(X1,matrix(0,zero.row(tr.n1),ncol(X0)))
W.sk.hada <- ((t(hada.sk.x0.und)%*%hada.sk.x0.und)/
                  (tr.n0-1)+ (t(hada.sk.x1.und)%*%
                      hada.sk.x1.und)/(tr.n1-1))/2

a.sk.hada <- t(solve(W.sk.hada,tol=1e-45)%*%
                   (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.sk.hada)

# Hadamard Sketching classification
cl.sk.hada <- ifelse(train.x%*%t(a.sk.hada) >
   c(1/2*a.sk.hada%*%(mean.tr.x0+mean.tr.x1)),1,0)

test.sk.hada <- test.x%*%t(a.sk.hada)
test.cl.sk.hada <- ifelse(test.sk.hada >
        c(1/2*a.sk.hada%*%(colMeans(train.x0) +
                     colMeans(train.x1))),1,0)

# Hadamard oversketching
hada.sk0.ov <- hada.sk(n=tr.n0,k=tr.n0)
hada.sk1.ov <- hada.sk(n=tr.n1,k=tr.n0)

hada.sk.x0.ov <- hada.sk0.ov%*%
```

```
      rbind(X0,matrix(0,zero.row(tr.n0),ncol(X0)))
hada.sk.x1.ov <- hada.sk1.ov%*%
      rbind(X1,matrix(0,zero.row(tr.n1),ncol(X0)))
hada.osk <- rbind(hada.sk.x0.ov,hada.sk.x1.ov)
W.osk.hada <- ((t(hada.sk.x0.ov)%*%hada.sk.x0.ov)/
     (tr.n0-1)+(t(hada.sk.x1.ov)%*%hada.sk.x1.ov)/
                                      (tr.n1-1))/2

a.osk.hada <- t(solve(W.osk.hada,tol=1e-45)%*%
                          (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.osk.hada)

# Hadamard Over-Sketching classification
cl.osk.hada <- ifelse(train.x%*%t(a.osk.hada) >
  c(1/2*a.osk.hada %*%(mean.tr.x0+mean.tr.x1)),1,0)
test.osk.hada <- test.x%*%t(a.osk.hada)
test.cl.osk.hada <- ifelse(test.osk.hada >
          c(1/2*a.osk.hada%*%(colMeans(train.x0) +
                         colMeans(train.x1))),1,0)

# Balanced Hadamard sketching + oversketching k=2*n1
hada.sk0.bal <- hada.sk(n=tr.n0,k=2*tr.n1)
hada.sk1.bal <- hada.sk(n=tr.n1,k=2*tr.n1)

hada.sk.x0.bal <-hada.sk0.bal%*%
        rbind(X0,matrix(0,zero.row(tr.n0),ncol(X0)))
hada.sk.x1.bal <- hada.sk1.bal%*%
        rbind(X1,matrix(0,zero.row(tr.n1),ncol(X0)))
hada.bal <- rbind(hada.sk.x0.bal,hada.sk.x1.bal)
W.bal.hada <- ((t(hada.sk.x0.bal)%*%hada.sk.x0.bal)/
     (tr.n0-1)+(t(hada.sk.x1.bal)%*%hada.sk.x1.bal)/
                                        (tr.n1-1))/2

a.bal.hada <- t(solve(W.bal.hada,tol=1e-45)%*%
                            (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.bal.hada)


# Hadamard Balanced Sketching classification
cl.bal.hada <- ifelse(train.x%*%t(a.bal.hada) >
    c(1/2*a.bal.hada%*%(mean.tr.x0+mean.tr.x1)),1,0)
```

```
test.bal.hada <- test.x%*%t(a.bal.hada)
test.cl.bal.hada <- ifelse(test.bal.hada >
            c(1/2*a.bal.hada%*%(colMeans(train.x0) +
                            colMeans(train.x1))),1,0)


# Gaussian Sketching ####
gauss.sk0.und <- gauss.sk(n=tr.n0,k=tr.n1)
gauss.sk1.und <- gauss.sk(n=tr.n1,k=tr.n1)

gauss.sk.x0.und <- gauss.sk0.und%*%X0
gauss.sk.x1.und <-gauss.sk1.und%*%X1
W.sk.gauss <- ((t(gauss.sk.x0.und)%*%gauss.sk.x0.und)/
   (tr.n0-1) + (t(gauss.sk.x1.und)%*%gauss.sk.x1.und)/
                                          (tr.n1-1))/2

a.sk.gauss <- t(solve(W.sk.gauss,tol=1e-45)%*%
                              (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.sk.gauss)

# Gaussian Sketching classification
cl.sk.gauss <- ifelse(train.x%*%t(a.sk.gauss)>
       c(1/2*a.sk.gauss%*%(mean.tr.x0+mean.tr.x1)),1,0)

test.sk.gauss <- test.x%*%t(a.sk.gauss)
test.cl.sk.gauss <- ifelse(test.sk.gauss >
             c(1/2*a.sk.gauss%*%(colMeans(train.x0) +
                             colMeans(train.x1))),1,0)

# Gaussian Over-Sketching
gauss.sk0.ov <- gauss.sk(n=tr.n0,k=tr.n0)
gauss.sk1.ov <- gauss.sk(n=tr.n1,k=tr.n0)

gauss.sk.x0.ov <- gauss.sk0.ov%*%X0
gauss.sk.x1.ov <- gauss.sk1.ov%*%X1
gauss.osk <- rbind(gauss.sk.x0.ov,gauss.sk.x1.ov)
W.osk.gauss <- ((t(gauss.sk.x0.ov)%*%gauss.sk.x0.ov)/
      (tr.n0-1)+(t(gauss.sk.x1.ov)%*%gauss.sk.x1.ov)/
                                         (tr.n1-1))/2

a.osk.gauss <- t(solve(W.osk.gauss,tol=1e-45)%*%
                             (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.osk.gauss)

# Gaussian Over-Sketching classification
cl.osk.gauss <- ifelse(train.x%*%t(a.osk.gauss) >
    c(1/2*a.osk.gauss%*%(mean.tr.x0+mean.tr.x1)),1,0)
test.osk.gauss <- test.x%*%t(a.osk.gauss)
test.cl.osk.gauss <- ifelse(test.osk.gauss >
    c(1/2*a.osk.gauss%*%(colMeans(train.x0) +
                    colMeans(train.x1))),1,0)

# Balanced Gaussian Sketching + Over-Sketching k=2*n1
gauss.sk0.bal <- gauss.sk(n=tr.n0,k=2*tr.n1)
gauss.sk1.bal <- gauss.sk(n=tr.n1,k=2*tr.n1)

gauss.sk.x0.bal <- gauss.sk0.bal%*%X0
gauss.sk.x1.bal <- gauss.sk1.bal%*%X1
gauss.bal <- rbind(gauss.sk.x0.bal,gauss.sk.x1.bal)
W.bal.gauss <- ((t(gauss.sk.x0.bal)%*%gauss.sk.x0.bal)/
    (tr.n0-1) + (t(gauss.sk.x1.bal)%*%gauss.sk.x1.bal)/
                                          (tr.n1-1))/2

a.bal.gauss <- t(solve(W.bal.gauss,tol=1e-45)%*%
                        (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.bal.gauss)

# Gaussian Balanced Sketching classification
cl.bal.gauss <- ifelse(train.x%*%t(a.bal.gauss) >
        c(1/2*a.bal.gauss%*%
                    (mean.tr.x0+mean.tr.x1)),1,0)
test.bal.gauss <- test.x%*%t(a.bal.gauss)
test.cl.bal.gauss <- ifelse(test.bal.gauss >
        c(1/2*a.bal.gauss%*%(colMeans(train.x0) +
                        colMeans(train.x1))),1,0)


# Clarkson-Woodruff Sketching ####
cw.sk0.und <- cw.sk(n=tr.n0,k=tr.n1)
cw.sk1.und <- cw.sk(n=tr.n1,k=tr.n1)

cw.sk.x0.und <- cw.sk0.und%*%X0
cw.sk.x1.und <- cw.sk1.und%*%X1
W.sk.cw <- ((t(cw.sk.x0.und)%*%cw.sk.x0.und)/
  (tr.n0-1)+(t(cw.sk.x1.und)%*%cw.sk.x1.und)/
                                 (tr.n1-1))/2

a.sk.cw <- t(solve(W.sk.cw,tol=1e-45)%*%
                 (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.sk.cw)

# Sketching classification
cl.sk.cw <- ifelse(train.x%*%t(a.sk.cw) >
            c(1/2*a.sk.cw%*%(mean.tr.x0 +
                           mean.tr.x1)),1,0)

test.sk.cw <- test.x%*%t(a.sk.cw)
test.cl.sk.cw <- ifelse(test.sk.cw >
      c(1/2*a.sk.cw%*%(colMeans(train.x0) +
                   colMeans(train.x1))),1,0)

# Clarkson-Woodruff oversketching
cw.sk0.ov <- cw.sk(n=tr.n0,k=tr.n0)
cw.sk1.ov <- cw.sk(n=tr.n1,k=tr.n0)

cw.sk.x0.ov <- cw.sk0.ov%*%X0
cw.sk.x1.ov <- cw.sk1.ov%*%X1
cw.osk <- rbind(cw.sk.x0.ov,cw.sk.x1.ov)
W.osk.cw <- ((t(cw.sk.x0.ov)%*%cw.sk.x0.ov)/
   (tr.n0-1)+(t(cw.sk.x1.ov)%*%cw.sk.x1.ov)/
                                (tr.n1-1))/2

a.osk.cw <- t(solve(W.osk.cw,tol=1e-45)%*%
                    (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.osk.cw)

# Clarkson-Woodruff Over-Sketching classification
cl.osk.cw <- ifelse(train.x%*%t(a.osk.cw) >
                c(1/2*a.osk.cw%*%(mean.tr.x0 +
                             mean.tr.x1)),1,0)
test.osk.cw <- test.x%*%t(a.osk.cw)
test.cl.osk.cw <- ifelse(test.osk.cw >
        c(1/2*a.osk.cw%*%(colMeans(train.x0) +
                     colMeans(train.x1))),1,0)

# Balanced Clarkson-Woodruff Sketching +
#                                Oversketching k=2*n1
cw.sk0.bal <- cw.sk(n=tr.n0,k=2*tr.n1)
cw.sk1.bal <- cw.sk(n=tr.n1,k=2*tr.n1)

cw.sk.x0.bal <- cw.sk0.bal%*%X0
cw.sk.x1.bal <-cw.sk1.bal%*%X1
cw.bal <- rbind(cw.sk.x0.bal,cw.sk.x1.bal)
W.bal.cw <- ((t(cw.sk.x0.bal)%*%cw.sk.x0.bal)/
   (tr.n0-1)+(t(cw.sk.x1.bal)%*%cw.sk.x1.bal)/
                                  (tr.n1-1))/2

a.bal.cw <- t(solve(W.bal.cw,tol=1e-45)%*%
                      (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.bal.cw)

# Clarkson-Woodruff Balanced Sketching classification
cl.bal.cw <- ifelse(train.x%*%t(a.bal.cw) >
        c(1/2*a.bal.cw%*%(mean.tr.x0 +
                            mean.tr.x1)),1,0)
test.bal.cw <- test.x%*%t(a.bal.cw)
test.cl.bal.cw <- ifelse(test.bal.cw >
        c(1/2*a.bal.cw%*%(colMeans(train.x0) +
                     colMeans(train.x1))),1,0)


# Haar Sketching ####
haar.sk0.und <- haar.sk(n=tr.n0,k=tr.n1)
haar.sk1.und <- haar.sk(n=tr.n1,k=tr.n1)

haar.sk.x0.und <- haar.sk0.und%*%X0
haar.sk.x1.und <- haar.sk1.und%*%X1
W.sk.haar <- ((t(haar0.sk.x0.und)%*%haar.sk.x0.und)/
   (tr.n0-1) + (t(haar.sk.x1.und)%*%haar.sk.x1.und)/
                                        (tr.n1-1))/2

a.sk.haar <- t(solve(W.sk.haar,tol=1e-45)%*%
                    (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.sk.haar)
```

```
# Haar Sketching classification
cl.sk.haar <- ifelse(train.x%*%t(a.sk.haar) >
               c(1/2*a.sk.haar%*%(mean.tr.x0 +
                              mean.tr.x1)),1,0)

test.sk.haar <- test.x%*%t(a.sk.haar)
test.cl.sk.haar <- ifelse(test.sk.haar >
        c(1/2*a.sk.haar%*%(colMeans(train.x0) +
                       colMeans(train.x1))),1,0)

# Haar Over-Sketching
haar.sk0.ov <- haar.sk(n=tr.n0,k=tr.n0)
haar.sk1.ov <- haar.sk(n=tr.n1,k=tr.n0)

haar.sk.x0.ov <-haar.sk0.ov%*%X0
haar.sk.x1.ov <-haar.sk1.ov%*%X1
haar.osk <- rbind(haar.sk.x0.ov,haar.sk.x1.ov)
W.osk.haar <- ((t(haar.sk.x0.ov)%*%haar.sk.x0.ov)/
   (tr.n0-1) + (t(haar.sk.x1.ov)%*%haar.sk.x1.ov)/
                                      (tr.n1-1))/2

a.osk.haar <- t(solve(W.osk.haar,tol=1e-45)%*%
                          (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.osk.haar)

# Haar  Over-Sketching classification
cl.osk.haar <- ifelse(train.x%*%t(a.osk.haar) >
               c(1/2*a.osk.haar%*%(mean.tr.x0 +
                              mean.tr.x1)),1,0)
test.osk.haar <- test.x%*%t(a.osk.haar)
test.cl.osk.haar <- ifelse(test.osk.haar >
        c(1/2*a.osk.haar%*%(colMeans(train.x0) +
                        colMeans(train.x1))),1,0)

# Balanced Haar Sketching + Over-Sketching k=2*n1
haar.sk0.bal <- haar.sk(n=tr.n0,k=2*tr.n1)
haar.sk1.bal <- haar.sk(n=tr.n1,k=2*tr.n1)

haar.sk.x0.bal <- haar.sk0.bal%*%X0
haar.sk.x1.bal <- haar.sk1.bal%*%X1
haar.bal <- rbind(haar.sk.x0.bal,haar.sk.x1.bal)
W.bal.haar <- ((t(haar.sk.x0.bal)%*%haar.sk.x0.bal)/
   (tr.n0-1) + (t(haar.sk.x1.bal)%*%haar.sk.x1.bal)/
                                        (tr.n1-1))/2

a.bal.haar <- t(solve(W.bal.haar,tol=1e-45)%*%
                           (mean.tr.x1-mean.tr.x0))
proj.data <- train.x%*%t(a.bal.haar)

# Haar Balanced Sketching classification
cl.bal.haar <- ifelse(train.x%*%t(a.bal.haar) >
               c(1/2*a.bal.haar%*%(mean.tr.x0 +
                                mean.tr.x1)),1,0)
test.bal.haar <- test.x%*%t(a.bal.haar)
test.cl.bal.haar <- ifelse(test.bal.haar >
        c(1/2*a.bal.haar%*%(colMeans(train.x0) +
                        colMeans(train.x1))),1,0)
```

Classification with Trees

```
x <- as.matrix(x)
x <- x[c(which(y==0),which(y==1)),]
y <- y[c(which(y==0),which(y==1))]
x0 <- x[y==0,]; y0<-y[y==0];
x1 <- x[y==1,]; y1<-y[y==1];
n0 <- sum(y==0); n1<-sum(y==1); n<-n0+n1;
p <- ncol(x)

quali.te0 <- sample(1:n0,round(n0*0.25),replace=FALSE)
quali.te1 <- sample(1:n1,round(n1*0.25),replace=FALSE)

test.x0 <- x0[quali.te0,];  test.x1 <- x1[quali.te1,];
test.y0 <- y0[quali.te0];   test.y1 <- y1[quali.te1];
te.n0 <- nrow(test.x0); te.n1 <- nrow(test.x1);

test.x <- rbind(test.x0,test.x1)
test.y <- c(test.y0,test.y1)

train.x0 <- x0[-quali.te0,]; train.x1 <- x1[-quali.te1,];
train.y0 <- y0[-quali.te0];  train.y1 <- y1[-quali.te1];
tr.n0 <- nrow(train.x0); tr.n1 <- nrow(train.x1);
train.x <- rbind(train.x0,train.x1)
train.y <- c(train.y0,train.y1)

X0 <- scale(train.x0,T,F)
X1 <- scale(train.x1,T,F)
mean.tr.x0 <- colMeans(train.x0)
mean.tr.x1 <- colMeans(train.x1)
mean.te <- colMeans(test.x)
ntr <- nrow(train.x)


## Gaussian Group-wise Partial Sketching
g.sk0.und <- matrix(rnorm(prod(tr.n1,tr.n0),0,
        sd=1/sqrt(tr.n1)),tr.n1,tr.n0)
g.sk1.und <- matrix(rnorm(prod(tr.n1,tr.n1),0,
        sd=1/sqrt(tr.n1)),tr.n1,tr.n1)

sk.x0.und <- g.sk0.und%*%X0+
    matrix(mean.tr.x0,nrow(g.sk0.und),ncol=p,byrow=T)
sk.x1.und <- g.sk1.und%*%X1+
    matrix(mean.tr.x1,nrow(g.sk1.und),ncol=p,byrow=T)


x.gwps <- as.data.frame(rbind(sk.x0.und,sk.x1.und))
y.gwps <- c(rep(0,nrow(sk.x0.und)),rep(1,nrow(sk.x1.und)))
df.gwps <- as.data.frame(cbind(y.gwps,x.gwps))
df.gwps$y.gwps <- as.factor(y.gwps)

## Gaussian Group-wise Partial Over-Sketching
g.sk0.ov <- matrix(rnorm(prod(tr.n0,tr.n0),0,
                     sd=1/sqrt(tr.n0)),tr.n0,tr.n0)
g.sk1.ov <- matrix(rnorm(prod(tr.n0,tr.n1),0,
                     sd=1/sqrt(tr.n0)),tr.n0,tr.n1)

sk.x0.ov <- g.sk0.ov%*%X0 +
  matrix(mean.tr.x0,nrow(g.sk0.ov),ncol=p,byrow=T)
sk.x1.ov <- g.sk1.ov%*%X1 +
  matrix(mean.tr.x1,nrow(g.sk1.ov),ncol=p,byrow=T)

x.gwos <- as.data.frame(rbind(sk.x0.ov,sk.x1.ov))
y.gwos <- c(rep(0,nrow(sk.x0.ov)),
                             rep(1,nrow(sk.x1.ov)))
df.gwos <- as.data.frame(cbind(y.gwos,x.gwos))
df.gwos$y.gwos <- as.factor(y.gwos)


# Hadamard Sketching
hada.sk0.und <- hada.sk(tr.n0,tr.n1)
hada.sk1.und <- hada.sk(tr.n1,tr.n1)

hada.sk.x0.und <- hada.sk0.und%*%
  (rbind(X0,matrix(0,zero.row(tr.n0),ncol(X0)))) +
   matrix(mean.tr.x0,nrow(hada.sk0.und),ncol=p,
                                       byrow=T)
hada.sk.x1.und <- hada.sk1.und%*%
  (rbind(X1,matrix(0,zero.row(tr.n1),ncol(X0)))) +
  matrix(mean.tr.x1,nrow(hada.sk1.und),ncol=p,
                                       byrow=T)
```

```
x.hadawus <- as.data.frame(rbind(hada.sk.x0.und,
                                hada.sk.x1.und))
y.hadawus <- c(rep(0,nrow(hada.sk.x0.und)),
                    rep(1,nrow(hada.sk.x1.und)))
df.hadawus <- as.data.frame(cbind(y.hadawus,
                                     x.hadawus))
df.hadawus$y.hadawus <- as.factor(y.hadawus)


# Hadamard Over-Sketching
hada.sk0.ov <- hada.sk(n=tr.n0,k=tr.n0)
hada.sk1.ov <- hada.sk(n=tr.n1,k=tr.n0)

hada.sk.x0.ov <- hada.sk0.ov%*%
  (rbind(X0,matrix(0,zero.row(tr.n0),ncol(X0)))) +
  matrix(mean.tr.x0,nrow(hada.sk0.ov),ncol=p,
                                         byrow=T)
hada.sk.x1.ov <- hada.sk1.ov%*%
  (rbind(X1,matrix(0,zero.row(tr.n1),ncol(X0)))) +
  matrix(mean.tr.x1,nrow(hada.sk1.ov),ncol=p,
                                         byrow=T)

x.hadawos <- as.data.frame(rbind(hada.sk.x0.ov,
                                hada.sk.x1.ov))
y.hadawos <- c(rep(0,nrow(hada.sk.x0.ov)),
                    rep(1,nrow(hada.sk.x1.ov)))
df.hadawos <- as.data.frame(cbind(y.hadawos,
                                    x.hadawos))
df.hadawos$y.hadawos <- as.factor(y.hadawos)

# Gaussian Balanced Sketching + Over-Sketching k= 2*n1
g.sk0.bal <- matrix(rnorm(prod(2*tr.n1,tr.n0),0,
              sd=1/sqrt(2*tr.n1)),2*tr.n1,tr.n0)
g.sk1.bal <- matrix(rnorm(prod(2*tr.n1,tr.n1),0,
              sd=1/sqrt(1*tr.n1)),2*tr.n1,tr.n1)

sk.x0.bal <- g.sk0.bal%*%X0 +
    matrix(mean.tr.x0,nrow(g.sk0.bal),ncol=p,
                                        byrow=T)
sk.x1.bal <- g.sk1.bal%*%X1 +
    matrix(mean.tr.x1,nrow(g.sk1.bal),ncol=p,
                                        byrow=T)

x.gwbal <- as.data.frame(rbind(sk.x0.bal,
                                      sk.x1.bal))
y.gwbal <- c(rep(0,nrow(sk.x0.bal)),
                          rep(1,nrow(sk.x1.bal)))
df.gwbal <- as.data.frame(cbind(y.gwbal,x.gwbal))
df.gwbal$y.gwbal <- as.factor(y.gwbal)

# Hadamard Balanced Sketching + Over-Sketching k=2*n1
hada.sk0.bal <- hada.sk(tr.n0,2*tr.n1)
hada.sk1.bal <- hada.sk2(tr.n1,2*tr.n1)

hada.sk.x0.bal <- hada.sk0.bal%*%
  (rbind(X0,matrix(0,zero.row(tr.n0),ncol(X0)))) +
   matrix(mean.tr.x0,nrow(hada.sk0.bal),ncol=p,
                                          byrow=T)
hada.sk.x1.bal <- hada.sk1.bal%*%
  (rbind(X1,matrix(0,zero.row(tr.n1),ncol(X0))))+
   matrix(mean.tr.x1,nrow(hada.sk1.bal),ncol=p,
                                          byrow=T)

x.hadawbal <- as.data.frame(rbind(hada.sk.x0.bal,
                                  hada.sk.x1.bal))
y.hadawbal <- c(rep(0,nrow(hada.sk.x0.bal)),
                      rep(1,nrow(hada.sk.x1.bal)))
df.hadawbal <- as.data.frame(cbind(y.hadawbal,
                                      x.hadawbal))
df.hadawbal$y.hadawbal <- as.factor(y.hadawbal)


# Clarkson-Woodruff Sketching
clark.sk0.und <- clark.sk(n=tr.n0,k=tr.n1)
clark.sk1.und <- clark.sk(n=tr.n1,k=tr.n1)

clark.sk.x0.und <- clark.sk0.und%*%X0 +
   matrix(mean.tr.x0, nrow(clark.sk0.und), ncol=p,
                                          byrow=T)
clark.sk.x1.und <- clark.sk1.und%*%X1 +
   matrix(mean.tr.x1, nrow(clark.sk1.und), ncol=p,
                                          byrow=T)
```

```
x.clarkwus <- as.data.frame(rbind(clark.sk.x0.und,
                                  clark.sk.x1.und))
y.clarkwus <- c(rep(0,nrow(clark.sk.x0.und)),
                rep(1,nrow(clark.sk.x1.und)))
df.clarkwus <- as.data.frame(cbind(y.clarkwus,
                                   x.clarkwus))
df.clarkwus$y.clarkwus <- as.factor(y.clarkwus)


# Clarkson-Woodruff Over-Sketching
clark.sk0.ov <- clark.sk(n=tr.n0,k=tr.n0)
clark.sk1.ov <- clark.sk(n=tr.n1,k=tr.n0)

clark.sk.x0.ov <- clark.sk0.ov%*%X0 +
    matrix(mean.tr.x0, nrow(clark.sk0.ov), ncol=p,
                                           byrow=T)
clark.sk.x1.ov <- clark.sk1.ov%*%X1 +
    matrix(mean.tr.x1, nrow(clark.sk1.ov), ncol=p,
                                           byrow=T)

x.clarkwos <- as.data.frame(rbind(clark.sk.x0.ov,
                                   clark.sk.x1.ov))
y.clarkwos <- c(rep(0,nrow(clark.sk.x0.ov)),
                 rep(1,nrow(clark.sk.x1.ov)))
df.clarkwos <- as.data.frame(cbind(y.clarkwos,
                                     x.clarkwos))
df.clarkwos$y.clarkwos <- as.factor(y.clarkwos)


# Clarkson-Woodruff Balanced Sketching +
#                          Over-Sketching k=2*n1
clark.sk0.bal <- clark.sk(n=tr.n0,k=2*tr.n1)
clark.sk1.bal <- clark.sk(n=tr.n1,k=2*tr.n1)

clark.sk.x0.bal <- clark.sk0.bal%*%X0 +
    matrix(mean.tr.x0, nrow(clark.sk0.bal), ncol=p,
                                            byrow=T)
clark.sk.x1.bal <- clark.sk1.bal%*%X1 +
    matrix(mean.tr.x1, nrow(clark.sk1.bal), ncol=p,
                                            byrow=T)
```

```
x.clarkwbal <- as.data.frame(rbind(clark.sk.x0.bal,
                                   clark.sk.x1.bal))
y.clarkwbal <- c(rep(0,nrow(clark.sk.x0.bal)),
                 rep(1,nrow(clark.sk.x1.bal)))
df.clarkwbal <- as.data.frame(cbind(y.clarkwbal,
                                    x.clarkwbal))
df.clarkwbal$y.clarkwbal <- as.factor(y.clarkwbal)


# Tree ++ Training Set
tree.gwps <- J48(y.gwps~., data=df.gwps, control =
                   Weka_control(S = TRUE, M = 5))
train.gwps <- predict(tree.gwps,x.gwps)

tree.gwos <- J48(y.gwos~., data=df.gwos, control =
                   Weka_control(S = TRUE, M = 5))
train.gwos <- predict(tree.gwos,x.gwos)

tree.hadawus <- J48(y.hadawus~., data=df.hadawus,
              control = Weka_control(S = TRUE, M = 5))
train.hadawus <- predict(tree.hadawus,x.hadawus)

tree.hadawos <- J48(y.hadawos~., data=df.hadawos,
              control = Weka_control(S = TRUE, M = 5))
train.hadawos <- predict(tree.hadawos,x.hadawos)

tree.gwbal <- J48(y.gwbal~., data=df.gwbal,
              control = Weka_control(S = TRUE, M = 5))
train.gwbal <- predict(tree.gwbal,x.gwbal)

tree.hadawbal <- J48(y.hadawbal~., data=df.hadawbal,
              control = Weka_control(S = TRUE, M = 5))
train.hadawbal <- predict(tree.hadawbal,x.hadawbal)

tree.clarkwus <- J48(y.clarkwus~., data=df.clarkwus,
              control = Weka_control(S = TRUE, M = 5))
train.clarkwus <- predict(tree.clarkwus,x.clarkwus)

tree.clarkwos <- J48(y.clarkwos~., data=df.clarkwos,
              control = Weka_control(S = TRUE, M = 5))
```

```
train.clarkwos <- predict(tree.clarkwos,x.clarkwos)

tree.clarkwbal <- J48(y.clarkwbal˜., data=df.clarkwbal,
              control = Weka_control(S = TRUE, M = 5))
train.clarkwbal <- predict(tree.clarkwbal,x.clarkwbal)


# Test Set
test.gwps <- predict(tree.gwps,
                newdata=as.data.frame(test.x))

test.pos <- predict(tree.gwos,
                newdata=as.data.frame(test.x))

hada.test.gwps <- predict(tree.hadawus,
                newdata=as.data.frame(test.x))

hada.test.gwos <- predict(tree.hadawos,
                newdata=as.data.frame(test.x))

test.gwbal <- predict(tree.gwbal,
                newdata=as.data.frame(test.x))

hada.test.gwbal <- predict(tree.hadawbal,
                newdata=as.data.frame(test.x))

clark.test.gwps <- predict(tree.clarkwus,
                newdata=as.data.frame(test.x))

clark.test.gwos <- predict(tree.clarkwos,
                newdata=as.data.frame(test.x))

clark.test.gwbal <- predict(tree.clarkwbal,
                newdata=as.data.frame(test.x))
```

Bibliography

AHFOCK, DANIEL, ASTLE, WILLIAM J, & RICHARDSON, SYLVIA. 2017. Statistical properties of sketching algorithms. *arxiv preprint arxiv:1706.03665*.

AILON, NIR, & CHAZELLE, BERNARD. 2009. The fast Johnson–Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, **39**(1), 302–322.

ANDERSON, THEODORE WILBUR. 1958. *An introduction to Multivariate Statistical Analysis*. First edn. Wiley New York.

BLOCKI, JEREMIAH, BLUM, AVRIM, DATTA, ANUPAM, & SHEFFET, OR. 2012. The Johnson-Lindenstrauss transform itself preserves differential privacy. *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, 410–419.

CANNINGS, TIMOTHY I, & SAMWORTH, RICHARD J. 2017. Random-projection ensemble classification. *Journal of the Royal Statistical Society: Series B*, **79**(4), 959–1035.

CERIOLI, ANDREA, & PERROTTA, DOMENICO. 2014. Robust clustering around regression lines with high density regions. *Advances in Data Analysis and Classification*, **8**(1), 5–26.

CHAWLA, NITESH V, BOWYER, KEVIN W, HALL, LAWRENCE O, & KEGELMEYER, W PHILIP. 2002. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, **16**, 321–357.

CHAWLA, NITESH V, JAPKOWICZ, NATHALIE, & KOTCZ, ALEKSANDER. 2004. Editorial: Special issue on Learning from Imbalanced Data Sets. *ACM sigkdd explorations newsletter*, **6**(1), 1–6.

CLARKSON, KENNETH L, & WOODRUFF, DAVID P. 2017. Low-rank approximation and regression in input sparsity time. *Journal of the Association for Computing Machinery (JACM)*, **63**(6), 54.

DASGUPTA, SANJOY, & GUPTA, ANUPAM. 2003. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, **22**(1), 60–65.

DHILLON, PARAMVEER, LU, YICHAO, FOSTER, DEAN P, & UNGAR, LYLE. 2013. New subsampling algorithms for fast least squares regression. *Pages 360–368 of: Advances in Neural Information Processing Systems 26 (NIPS).*

DOBRIBAN, EDGAR, & LIU, SIFAN. 2018. A new theory for sketching in linear regression. *arxiv preprint arxiv:1810.06089.*

FERN, XIAOLI Z, & BRODLEY, CARLA E. 2003. Random projection for high dimensional data clustering: A cluster ensemble approach. *Pages 186–193 of: Proceedings of the 20th International Conference on Machine Learning (ICML-03).*

FISHER, RONALD A. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, **7**(2), 179–188.

FITHIAN, WILLIAM, & HASTIE, TREVOR. 2014. Local case-control sampling: Efficient subsampling in imbalanced data sets. *Annals of Statistics*, **42**(5), 1693–1724.

GATARIC, MILANA, WANG, TENGYAO, & SAMWORTH, RICHARD. 2017. Sparse Principal Component Analysis via Random Projections. *Journal of the Royal Statistical Society: Series B*, 12.

GUPTA, ARJUN K, & NAGAR, DAYA K. 2018. *Matrix variate distributions*. Chapman and Hall/CRC.

HAFF, LR. 1979. An identity for the Wishart distribution with applications. *Journal of Multivariate Analysis*, **9**(4), 531–544.

HORN, ROGER A., & JOHNSON, CHARLES R. 1991. *Topics in matrix analysis*. Cambridge: Cambridge University Press.

JOHNSON, WILLIAM B, & LINDENSTRAUSS, JORAM. 1984. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, **26**(189-206), 1.

LUNARDON, NICOLA, MENARDI, GIOVANNA, & TORELLI, NICOLA. 2014. ROSE: A Package for Binary Imbalanced Learning. *R journal*, **6**(1).

MAHONEY, MICHAEL W. 2011. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, **3**(2), 123–224.

MANI, INDERJEET, & ZHANG, I. 2003. *k*NN approach to unbalanced data distributions: a case study involving information extraction. *In: Proceedings of Workshop on Learning from Imbalanced Datasets*, vol. 126.

MARZETTA, THOMAS L, TUCCI, GABRIEL H, & SIMON, STEVEN H. 2011. A random matrix-theoretic approach to handling singular covariance estimates. *IEEE Transactions on Information Theory*, **57**(9), 6256–6271.

MCLACHLAN, GEOFFREY. 1992. *Discriminant analysis and statistical pattern recognition*. first edn. Vol. 235. John Wiley & Sons.

MENARDI, GIOVANNA, & TORELLI, NICOLA. 2014. Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, **28**(1), 92–122.

MILLER, KENNETH S. 1981. On the inverse of the sum of matrices. *Mathematics magazine*, **54**(2), 67–72.

PILANCI, MERT, & WAINWRIGHT, MARTIN J. 2016. Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares. *The Journal of Machine Learning Research*, **17**(1), 1842–1879.

QUINLAN, J ROSS. 2014. *C4. 5: Programs for Machine Learning*. Elsevier.

SARLÓS, TAMÁS. 2006. Improved approximation algorithms for large matrices via random projections. *Pages 143–152 of: 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06).*

THANEI, GIAN-ANDREA, HEINZE, CHRISTINA, & MEINSHAUSEN, NICOLAI. 2017. Random projections for large-scale regression. *Pages 51–68 of: Big and Complex Data Analysis*. Springer.

WOODRUFF, DAVID P. 2014. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, **10**(1–2), 1–157.

WOODS, KEVIN S., DOSS, CHRISTOPHER C., BOWYER, KEVIN W., SOLKA, JEFFREY L., PRIEBE, CAREY E., & KEGELMEYER, W. PHILIP. 1993. Comparative Evaluation of Pattern Recognition Techniques for Detection of Microcalcifications in Mammography. *International Journal of Pattern Recognition and Artificial Intelligence*, **07**(06), 1417–1436.

XIE, JIGANG, & QIU, ZHENGDING. 2007. The effect of imbalanced data sets on LDA: A theoretical and empirical analysis. *Pattern recognition*, **40**(2), 557–562.

XUE, JING-HAO, & HALL, PETER. 2014. Why does rebalancing class-unbalanced data improve AUC for linear discriminant analysis? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**(5), 1109–1112.

XUE, JING-HAO, & TITTERINGTON, D MICHAEL. 2008. Do unbalanced data have a negative effect on LDA? *Pattern recognition*, **41**(5), 1558–1571.