

# Fast feature selection via knockoffs

## Efficiently control the false discovery rate



Faculté Informatique et Communications  
École Polytechnique Fédérale de Lausanne  
Master thesis of

**Quentin Rebjock**

Supervised by

Prof. Alexandre d'Aspremont, Sierra team, Inria, ENS, CNRS  
Prof. Martin Jaggi, Machine Learning and Optimization lab, EPFL

Paris, Inria, March 2020



Wings are a constraint that makes  
it possible to fly.  
— Robert Bringhurst

To my parents...



# Acknowledgements

I would like to thank my thesis advisor Alexandre



# Abstract

**Abstract**





# Notations

$\mathbb{N}$  denotes the set of natural integers and  $\mathbb{R}$  the set of real numbers. Vectors are written in bold letters, e.g.  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$ . Matrices are in capital bold letters, e.g.  $\mathbf{X} \in \mathbb{R}^{n \times p}$ . We may cast any vector  $\mathbf{x} \in \mathbb{R}^p$  to a column matrix  $x \in \mathbb{R}^{p \times 1}$  with the same entry values. We note  $\odot$  the Hadamard (element-wise) product between two vectors or matrices.  $\mathbf{0}$  and  $\mathbf{1}$  are the vectors whose all entries are 0 and 1 respectively, and whose size is inferred by the context. For a vector  $\mathbf{v}$ , we note  $\|\mathbf{v}\|_0$  the number of non-zero entries of  $\mathbf{v}$  (that is, its cardinality). The  $\text{diag}$  operator can be used in two contexts; either to transform a vector into a diagonal matrix whose diagonal entries match the vector, or to extract the diagonal of a matrix. For example

$$\text{diag} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \quad \text{diag} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$$

Given two matrices  $A$  and  $B$ , we note  $[A, B]$  and  $\begin{bmatrix} A \\ B \end{bmatrix}$  their horizontal and vertical concatenation respectively (if dimensions match). Functions  $\mathbb{R} \rightarrow \mathbb{R}$  applied to matrices are implicitly performed element-wise, unless otherwise specified.  $\mathbb{E} \mathcal{X}$  denotes the expectation of a random variable (or vector)  $\mathcal{X}$ .

Given a set  $\mathcal{S}$ ,  $|\mathcal{S}|$  and  $\#\mathcal{S}$  denote its cardinality.



# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Notations</b>	<b>v</b>
<b>Introduction</b>	<b>1</b>
<b>1 Feature selection</b>	<b>3</b>
1.1 Background on feature selection . . . . .	3
1.1.1 Definitions . . . . .	3
1.1.2 Motivation . . . . .	3
1.1.3 Techniques . . . . .	4
Lasso . . . . .	4
Sparse center classifiers . . . . .	4
1.2 False discovery rate control . . . . .	4
1.2.1 Definitions . . . . .	4
1.2.2 Benjamini–Hochberg–Yekutieli procedures . . . . .	5
Benjamini–Hochberg . . . . .	5
Benjamini–Yekutieli . . . . .	5
1.3 FDR control with knockoffs . . . . .	6
1.3.1 Knockoffs construction . . . . .	6
General case . . . . .	7
Gaussian case . . . . .	7
1.3.2 Statistics computation . . . . .	8
General principle . . . . .	8
Statistics aggregation . . . . .	8
Examples . . . . .	8
1.3.3 Selection thresholds . . . . .	9
1.3.4 Bottlenecks . . . . .	9
SDP . . . . .	9
Statistics computation . . . . .	10
1.3.5 Python implementation . . . . .	10
<b>2 Sparse naive Bayes</b>	<b>11</b>
2.1 Reminders on vanilla naive Bayes . . . . .	11
2.1.1 General settings . . . . .	11
2.1.2 Gaussian naive Bayes . . . . .	12
2.1.3 Bernoulli naive Bayes . . . . .	12
2.1.4 Multinomial naive Bayes . . . . .	12
2.1.5 Decision boundary . . . . .	13
2.2 Sparse naive Bayes . . . . .	13
2.2.1 Problem statement . . . . .	14
2.2.2 Main results and resolution . . . . .	14
2.3 Applications . . . . .	15
2.3.1 Criteo dataset . . . . .	15
<b>3 Fast statistics computation</b>	<b>17</b>
3.1 Lasso is good but slow . . . . .	17
3.2 Fast statistics . . . . .	17
3.3 Multi-stage procedures . . . . .	17

<b>4</b>	<b>SDP</b>	<b>19</b>
4.1	Equi-correlated knockoffs . . . . .	19
	A cheap solution . . . . .	19
	Why it is not desirable . . . . .	19
4.2	SDP knockoffs . . . . .	19
<b>A</b>	<b>An appendix</b>	<b>21</b>

# Introduction

Feature selection In the past two decades, acquiring data has become Datasets end up with a number of features of several dozens of thousands. Often, it is also easier to measure features than to label the sample with the right category. Indeed, the latter often requires a human and is very costly. in this setup, feature selection is particularly challenging as the number of observed samples might be much smaller than the number of covariates. It is for example the case of fMRI data and genomic data.

In this master thesis, The first chapter introduces the concept of feature selection, false discovery rate, and the knockoff framework developed by Barber-Candès. Chapter 2 introduces a sparse version of naive Bayes that scales linearly in the number of features against which it is trained.



## Chapter 1

# Feature selection

We introduce in this chapter the concepts of feature selection and false discovery rate control, as well as a short review of a few methods that are usually employed. Finally, we detail the knockoff framework developed by Barber-Candès, a recent approach to control the false discovery rate when performing feature selection.

## 1.1 Background on feature selection

### 1.1.1 Definitions

Feature selection is an active area of research in statistics and machine learning. It primarily consists in identifying the most relevant features (or covariates) explaining an observed variable. It is closely related to Occam's razor which states that the simplest explanation is often the best one. More formally, let  $\mathcal{X}$  be a  $p$ -dimensional random vector representing observed features and  $\mathcal{Y}$  a random target that may depend on  $\mathcal{X}$ . For example,  $\mathcal{X}$  could be the level of expression of the genes of an individual, and  $\mathcal{Y} \in \{0, 1\}$  a binary response indicating whether or not the person has some disease. We wish to find the subset of covariates  $\mathcal{S} \subseteq \{1, \dots, p\}$  that best explains the target  $\mathcal{Y}$ , be it a regression or a classification problem. To do so, suppose that the joint vector  $(\mathcal{X}, \mathcal{Y})$  follows some distribution, that is  $(\mathcal{X}, \mathcal{Y}) \sim \mathcal{P}_{\mathcal{X}, \mathcal{Y}}$ . Even though finding the conditional distribution  $\mathcal{P}_{\mathcal{Y}|\mathcal{X}}$  is beyond hope, one may be interested in finding on which subset  $\mathcal{S}$  of features  $\mathcal{P}_{\mathcal{Y}|\mathcal{X}}$  depends. Such a subset is not necessarily unique Markov blanket [23], b [1]. We define the set of null features  $\mathcal{H}_0 \subseteq \{1, \dots, p\}$  as follows:  $j \in \mathcal{H}_0$  if and only if  $\mathcal{Y} \perp\!\!\!\perp \mathcal{X}_j \mid \mathcal{X}_{-j}$  (where  $\mathcal{X}_{-j}$  denotes that all column entries are kept except the  $j$ th). The goal is thus to find a procedure selection a subset of features  $\hat{\mathcal{S}} \subseteq \{1, \dots, p\}$ .

In practice, one would observe several independent realizations of  $(\mathcal{X}, \mathcal{Y})$  and would aggregate them into a feature matrix  $X \in \mathbb{R}^{n \times p}$  of  $n$  samples and  $p$  features, and a target vector  $\mathbf{y} \in \mathbb{R}^n$  respectively. The independence of the observations is a credible assumption in many real life settings

### 1.1.2 Motivation

Feature selection may be used in a multitude of contexts and we point out here the main reasons why one would want to perform it on a dataset.

1. Making the model more interpretable. It gives insights on the most relevant features to explain the observed target.
2. Facilitating data visualization. As only a small subset of the features is selected, projecting it to a low (2 or 3) dimensional space is easier.
3. Reducing the training time. Many machine learning algorithm have a super-linear time complexity in the number of features. Pre-selecting a small subsets of those with a cheaper method can noticeably as less[6]
4. Improving the generalization of machine learning models. Irrelevant features may only be noise and fitting a model against them at train time is prone to over-fitting noise.
5. Avoiding the curse of dimensionality [29]. For example, the  $k$ -nearest neighbors algorithms [2] is known to perform badly as the feature space dimension increases. The number of samples actually has to grow exponentially in the number of features in order for the algorithm to perform decently.

Recently, the cost of measuring more features has drastically decreased. Many datasets, and especially in biology, end up with several dozens of thousands of features. Most of these features are expected to be insignificant, but there is a priori no reason to eliminate them. Adding them to the feature matrix is cheap, and it is then the role of the machine learning algorithm to detect the relevant ones.

### 1.1.3 Techniques

A lot of different paradigms and techniques to perform feature selection exist [15] a [13]. b[16] c[10] d[27]

#### Lasso

First rediscovered by Tibshirani [28] in 1996, the Lasso has become increasingly popular because of its capacity to both shrink the coefficients towards 0 and to select a subset of the features. It is basically a linear least squares regression whose weights are penalized by their  $\ell_1$  norm, multiplied by some factor  $\lambda > 0$ .

$$\hat{\beta}(\lambda) = \arg \min_{\mathbf{b}} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_1 \quad (1.1)$$

The  $\ell_1$  penalty tends to make the weights  $\hat{\beta}(\lambda)$  sparse as  $\lambda$  increases. Actually, for any feature  $j$ , there is a  $\lambda_{\min}$  such that for all  $\lambda \geq \lambda_{\min}$ ,  $\hat{\beta}_j(\lambda) = 0$ . That wouldn't be the case with an  $\ell_2$  penalty, for which the coefficients tend to 0 without reaching that value. This property makes the Lasso particularly suited for feature selection; just keep the features whose weight is non-zero. However, the choice of  $\lambda$  seems to be arbitrary, especially as it's not possible to know beforehand how many features will be selected.

The behavior of the path  $\lambda \mapsto \hat{\beta}(\lambda)$  has been extensively studied, and the algorithm LARS [12] was developed to efficiently compute it, which is possible because it is piecewise linear. It allows to compute  $\hat{\beta}$  for all relevant value of  $\lambda$  at a marginal cost. In practice, the  $\lambda$  giving the highest score on a *train* dataset is picked. The  $\ell_1$  regularization can easily be extended to many other estimators than the least squares, as for example logistic regression or SVMs.

#### Sparse center classifiers

Nearest centroid classification [17] is a very simple classification scheme. It consists in computing the averages  $\theta^\pm = \sum_{j \in \mathcal{I}^\pm} \mathbf{x}_j \in \mathbb{R}^p$  of the data points from the positive and negative classes  $\mathcal{C}^\pm$ , where  $\mathcal{I}^\pm$  contain the positive and the negative points. A new data point  $\mathbf{x} \in \mathbb{R}^p$  is classified positive or negative depending on its closest average  $\theta^+$  or  $\theta^-$ . Finding the class averages can be formulated as the following optimization problem.

$$\arg \min_{\theta^+, \theta^-} \frac{1}{n^+} \sum_{i \in \mathcal{I}^+} \|\mathbf{x}_i - \theta^+\|_2^2 + \frac{1}{n^-} \sum_{i \in \mathcal{I}^-} \|\mathbf{x}_i - \theta^-\|_2^2 \quad (1.2)$$

Note  $\Delta$  the decision boundary, such that  $\mathbf{x}$  is classified  $\mathcal{C}^+$  if  $\Delta(\mathbf{x}) > 0$  and  $\mathcal{C}^-$  otherwise.

$$\Delta(\mathbf{x}) = \|\mathbf{x} - \theta^-\|_2^2 - \|\mathbf{x} - \theta^+\|_2^2 \quad (1.3)$$

The expression 1.3 can be expanded into  $\Delta(\mathbf{x}) = \|\theta^+\|_2^2 + \|\theta^-\|_2^2 + 2\mathbf{x}^\top(\theta^+ - \theta^-)$ , making it clear that the decision depends on the feature  $j$  if and only if  $\theta_j^+ \neq \theta_j^-$ . Using this observation, [8] introduced a  $\ell_0$  penalized version of 1.2 that can be solved very efficiently in closed-form. The authors obtain accuracy scores on par with the ones of the Lasso on the datasets they experiment. This a sparse centroid classification allows in particular to perform feature selection by keeping the features  $j$  such that  $\theta_j^+ \neq \theta_j^-$ .

None of the selection scheme presented above offer strong guarantees regarding the number of false positives that are detected. The selected subset  $\hat{\mathcal{S}}$  could potentially be disjoint to  $\mathcal{S}$  if the selection criterion is not adapted to the problem.

## 1.2 False discovery rate control

When performing feature selection one is usually interested in two quantities, namely the *false discovery rate* (FDR) and the *power*. Intuitively, the former (resp. the latter) assesses the expected proportion of false discoveries (resp. true discoveries) of a selection procedure.

### 1.2.1 Definitions

Suppose the conditional probability distribution  $\mathcal{Y} \mid \mathcal{X}$  depends only on a subset of features  $\mathcal{S} \subset \{1, \dots, p\}$ . A feature selection algorithm outputs a subset  $\hat{\mathcal{S}} \subset \{1, \dots, p\}$  (which is potentially random) that it judges to be relevant. The false discovery proportion (FDP), the false discovery rate (FDR), and the power are defined as follows

$$\text{FDP} = \frac{|\{\hat{\mathcal{S}} \setminus \mathcal{S}\}|}{|\hat{\mathcal{S}}|}, \quad \text{FDR} = \mathbb{E}[\text{FDP}], \quad \text{power} = \mathbb{E} \frac{|\{\hat{\mathcal{S}} \cap \mathcal{S}\}|}{|\hat{\mathcal{S}}|} \quad (1.4)$$



The FDP merely measures the proportion of features in  $\hat{\mathcal{S}}$  that are not in  $\mathcal{S}$ . It depends on the samples  $X$  and  $\mathbf{y}$ , and possibly on the inherent randomness of the selection algorithm. That is why the FDR assesses the expectation of this value. Finally, the power is the fraction of the important features that were actually discovered, in average.

Even though it is beyond hope to retrieve the whole set  $\mathcal{S}$  with no error, a multitude of techniques attempt to find as many relevant features as possible (that is, maximizing the power) while maintaining the FDR under a certain threshold. The concept of FDR was first introduced by [4] in 1995 and has become increasingly decisive in some sciences such as biology where the acquisition of a huge number of covariates is frequent. It is now possible to measure the expression of several dozens of thousands of genes for a given individual.

### 1.2.2 Benjamini–Hochberg–Yekutieli procedures

The Benjamini–Hochberg [4] and Benjamini–Yekutieli [5] schemes are two methods controlling the FDR that are widely used in practice. They are closely related to each other but offer different guarantees regarding the effective control of the FDR. BH is less conservative than BY but makes stronger assumptions on the  $p$ -values it manipulates.

For each feature  $j \in \{1, \dots, p\}$ , let  $\mathcal{H}_j$  be the null hypothesis ( $j$  does not belong to  $c\mathcal{S}$ ), and  $p_j$  the corresponding  $p$ -value. Let  $q \in [0, 1]$  be some FDR target that should not be exceeded. We note  $(p_{(j)})_j$  the sequence of  $p$ -values in increasing order;  $p_{(1)} \leq \dots \leq p_{(p)}$ . Let  $m_0$  be the number of true null hypotheses.

#### Benjamini–Hochberg

The Benjamini–Hochberg (BH) procedure was the first method proposed to control the FDR. It consists in the following steps:

1. Sorting the  $p$ -values such that  $p_{(1)} \leq \dots \leq p_{(p)}$
2. Finding the largest  $k \in \mathbb{N}$  such that  $p_{(k)} \leq \frac{k \cdot q}{p}$
3. Rejecting all the hypotheses  $\mathcal{H}_{(j)}$  such that  $j \in \{1, \dots, k\}$

In the end, all the features  $j$  such that  $p_j \leq p_{(k)}$  are selected. It guarantees that  $\text{FDR} \leq \frac{m_0}{p} q$  under the assumption that the  $p$ -values were computed independently, which can be very restrictive in many situations.

#### Benjamini–Yekutieli

Similarly, the Benjamini–Yekutieli procedure controls the FDR and does not need the  $p$ -values independence assumption, at the cost of a more conservative selection. It follows the same scheme, namely: ordering the  $p$ -values, finding the largest  $k \in \mathbb{N}$  such that  $p_{(k)} \leq \frac{k}{p \cdot c(p)} q$ , and rejecting  $\mathcal{H}_{(j)}$  if  $j \leq k$ . Note the additional factor  $c(p) \geq 1$ , defined in 1.5.

$$c(p) = \sum_{j=1}^p \frac{1}{j}, \quad \text{FDR} \leq \frac{m_0}{p} q \quad (1.5)$$

Both procedures are illustrated in figure 1.2. They require  $p$ -values to be computed, which is not always feasible, especially when  $p > n$ . Furthermore, BH and BY have stronger guarantees than desired; for a desired FDR level  $q$ , they ensure that  $\text{FDR} \leq \frac{m_0}{p} q$ , and the factor  $\frac{m_0}{p}$  might be much smaller than 1.

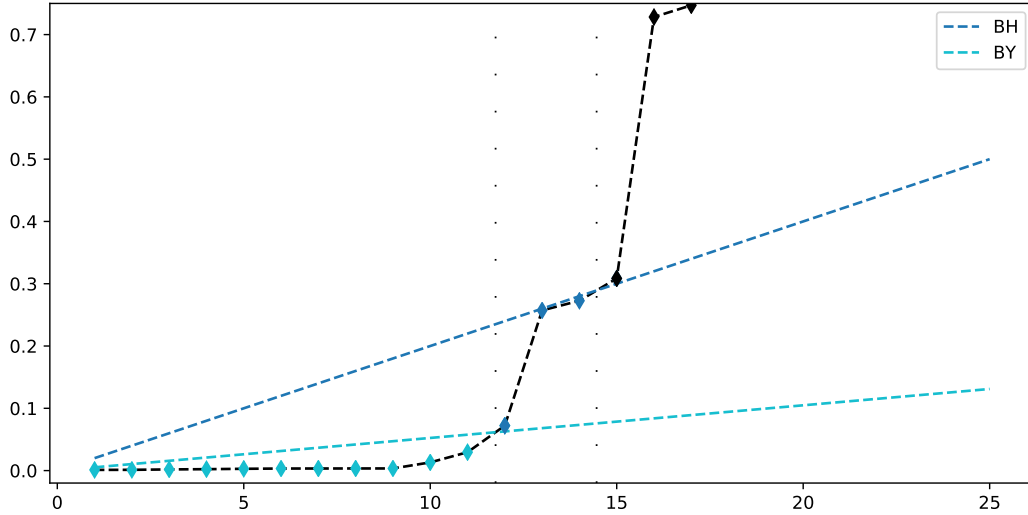


FIGURE 1.2: Illustration of the BH and the BY procedures. They are geometrically equivalent to drawing a line of slope  $\beta$  going through the origin, identifying the last  $p$ -value under the line, and keeping the features on the left of the intersection point. For BH,  $\beta = \frac{q}{p}$  (in blue), while for BY,  $\beta = \frac{q}{p \cdot c(p)}$  (in cyan). In this toy example,  $q = 0.5$ , and it shows that BY is way more conservative than BH.

### 1.3 FDR control with knockoffs

In 2015 Barber-Candès introduce the knockoffs framework and extend it later in 2018 to more general settings (allowing in particular to perform feature selection in the high dimension context). The goal is to control the false discovery rate as defined in 1.4, while maintaining a reasonable power. More formally, let  $X \in \mathbb{R}^{n \times p}$  be a feature matrix and  $\mathbf{y} \in \mathbb{R}^n$  the associated target vector. Suppose  $\mathbf{y}$  depends only on the subset of features  $\mathcal{S} \subseteq \{1, \dots, p\}$ . Given some FDR target  $q \in [0, 1]$ , we wish to find a procedure such as, in average, the false discovery proportion is smaller than  $q$ , i.e.  $\text{FDR} \leq q$ . To do so, the key idea is to construct for each original feature  $X_j$ ,  $j \in \{1, \dots, p\}$ , a knockoff (that is to say, fake) feature  $\tilde{X}_j$  which is known to be out of the model. Original features are then selected only if they prove to be more significant than their knockoff counterparts.

Note  $\Sigma = X^\top X$ .

To compare a feature and its knockoff several quantities will be computed by interchanging them. For this reason, we define in 1 the swap operator.

**Definition 1.** (swap operator) Given two matrices  $A, B \in \mathbb{R}^{n \times p}$  of same size, we define the swap operator on the concatenated matrix  $[A, B]$  as follows. For a any subset of indices  $S \subseteq \{1, \dots, p\}$ ,  $[A, B]_{\text{swap}(S)}$  is the transformed matrix where the columns  $A_j$  and  $B_j$  were swapped for all  $j \in S$ .

In the following sections, we are going to detail principal aspects of the construction of the knockoff features, the computation of statistics for both original and knockoff features, and the feature selection itself, based on those statistics.

#### 1.3.1 Knockoffs construction

In 2015, Barber-Candès introduced first the *fixed-X* knockoffs variable selection procedure. It relies on the creation of fake features satisfying some correlation constraints with the original features. Unfortunately, it can only perform adequately when  $n \geq 2p$ , even though it can be partially extended to the cases where  $n \geq p$ . In 2018, Candès proposed an extension of the framework called *model-X* knockoffs, in which fake features are sampled from a learned distribution. Despite the restriction of *fixed-X* knockoffs, that method is still appealing as the construction of the knockoff variables is straightforward (but costly). On the other hand, *model-X* knockoffs work in higher feature dimensions but need an estimation of the distribution that generated the data, which is a hard problem in general. We will focus on the construction of *model-X* knockoffs as we are principally interested in the high dimensional setting.

In this section, let  $(\mathcal{X}, \mathcal{Y})$  be a pair of random variables,  $\mathcal{X}$  being composed of  $p$  features, and  $\mathcal{Y}$  the associated target in  $\mathbb{R}$ . We suppose that the feature matrix and the target vector  $(X, \mathbf{y})$  are composed of samples from  $(\mathcal{X}, \mathcal{Y})$ . We wish to build knockoff features  $\tilde{X} \in \mathbb{R}^{n \times p}$ , and to do so we are going to sample from a random variable  $\tilde{\mathcal{X}}$  built such that  $[\mathcal{X}; \tilde{\mathcal{X}}]$  follows the properties [S.1](#) and [S.2](#) defined thereafter.

### General case

**Definition 2.** (*model-X knockoffs*) Given random vector  $\mathcal{X} \in \mathbb{R}^p$  of features, a random vector  $\tilde{\mathcal{X}} \in \mathbb{R}^p$  is said to be model- $\mathcal{X}$  knockoffs with respect to  $\mathcal{Y}$  if it satisfies the two following properties:

**S.1** For any  $S \subseteq \{1, \dots, p\}$ ,  $[\mathcal{X}; \tilde{\mathcal{X}}]_{\text{swap}(S)}^\top \stackrel{d}{=} [\mathcal{X}; \tilde{\mathcal{X}}]^\top$

**S.2**  $\tilde{\mathcal{X}} \perp\!\!\!\perp \mathcal{Y} \mid \mathcal{X}$

Intuitively, the condition [S.1](#) ensures that a knockoff feature is sufficiently close to its associated original feature so that swapping them doesn't change the distribution of the concatenated random vector. The independence condition [S.2](#) states that the knockoff features carry no additional information on  $\mathcal{Y}$ , given  $\mathcal{X}$ . It is trivially satisfied if  $\tilde{X}$  is built without exploiting  $\mathbf{y}$ . However, constructing knockoffs meeting the first distribution equality is practically infeasible in general.

**Remark 1.** Constructing the knockoffs feature matrix  $\tilde{X} \in \mathbb{R}^{n \times p}$ . It must satisfy two conditions, namely:

$$\tilde{X}^\top \tilde{X} = \Sigma,$$

$$X^\top \tilde{X} = \Sigma - \text{diag}\{\mathbf{s}\} \text{ where } \mathbf{s} \text{ is a non negative vector.}$$

It ensures that  $\tilde{X}$  has the same covariance as the original matrix  $X$  and that the correlation between distinct original and knockoff variables is the same as the correlation between the two originals.

### Gaussian case

In the particular case where  $\mathcal{X}$  is multivariate gaussian, the exact distribution of  $\tilde{\mathcal{X}}$  can be derived.

**Proposition 2.** Suppose that  $\mathcal{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ . If  $\tilde{\mathcal{X}}$  is a random variable such that

$$[\mathcal{X}; \tilde{\mathcal{X}}] \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix}, \Omega\right), \quad \text{where} \quad \Omega = \begin{bmatrix} \Sigma & \Sigma - \text{diag}\{\mathbf{s}\} \\ \Sigma - \text{diag}\{\mathbf{s}\} & \Sigma \end{bmatrix} \quad \text{for some } \mathbf{s} \in \mathbb{R}^p,$$

then  $[\mathcal{X}; \tilde{\mathcal{X}}]$  satisfies the swap property [S.1](#), provided that  $\Omega$  is positive semidefinite (so that it is indeed a covariance matrix).

In the proposition [1.3.1](#),  $\mathbf{s} \in \mathbb{R}^p$  can be any vector such that  $\Omega \succeq \mathbf{0}$ . We will come back later to the choice of  $\mathbf{s}$  which is actually crucial. For now, assume that  $\mathbf{s}$  satisfies this assumption. This result gives a way of constructing the knockoff features from  $X$ . Indeed, as  $[\mathcal{X}; \tilde{\mathcal{X}}]$  is multivariate normal, we may compute the exact distribution of  $\tilde{\mathcal{X}} \mid \mathcal{X}$  with classical formulas [\[11\]](#) as shown in [1.6](#).

$$\tilde{\mathcal{X}} \mid \mathcal{X} \sim \mathcal{N}(\mathbf{v}, \Upsilon), \quad \text{where} \quad \begin{cases} \mathbf{v} = \mathcal{X} - \mathcal{X}\Sigma^{-1} \text{diag}(\mathbf{s}) \\ \Upsilon = \text{diag}(\mathbf{s}) (2\mathbf{I}_{p \times p} - \Sigma^{-1} \text{diag}(\mathbf{s})) \end{cases} \quad (1.6)$$

To put this into practice, one would compute the empirical mean  $\hat{\boldsymbol{\mu}} \in \mathbb{R}^p$  and covariance  $\hat{\Sigma} \in \mathbb{R}^{p \times p}$  of  $\mathcal{X}$  using the observed feature matrix  $X$ . Then, each row of  $\tilde{X}$  is sampled according to a gaussian distribution  $\mathcal{N}(\mathbf{v}, \Upsilon)$  whose parameters are described in [1.6](#). Note in particular that the construction process is random; if it is repeated, it may very well return different knockoffs, and thus different selected features. Because of this instability, several attempts [\[26\]](#) to fix it by aggregating several knockoff samples.

Depending on the prior on the data and on the available computing power, several algorithms may be used to estimate the covariance matrix  $\Sigma$ . The empirical estimator is cheap but known to be unstable when  $p > n$ . Shrunk estimations like Ledoit-Wolf [\[18\]](#) provide good results.

The gaussian hypothesis is obviously rarely verified in practice but yields acceptable results even when  $\mathcal{X}$  is far from gaussian. It partly comes from the fact that, rather than constructing  $\tilde{\mathcal{X}}$  to respect [S.1](#), a weaker condition would be to enforce  $[\mathcal{X}; \tilde{\mathcal{X}}]$  and  $[\mathcal{X}; \tilde{\mathcal{X}}]_{\text{swap}(S)}$  to have the same first two moments (mean and covariance). It turns out to be the case if  $\tilde{\mathcal{X}} \mid \mathcal{X}$  is constructed as described in [1.6](#). In the remaining of this master thesis, we will restrain ourselves to the gaussian hypothesis.

The first knockoff paper [\[14\]](#) introducing the *fixed-X* knockoffs relied on similar correlation properties between the original and the knockoff features. It however imposed  $p \leq n$  and fewer statistics would yield theoretical guarantees regarding the FDR control of the procedure. In the reminding, we will only consider

*model-X* knockoffs as we are interested in the high dimensional setting. Moreover, *model-X* knockoffs tend to give higher power experimentally compared to their *fixed* counterparts.

Note that generating knockoffs requires an estimation of the distribution of  $\mathcal{X}$  only, and not  $\mathcal{Y} \mid \mathcal{X}$  as most methods would. It is particularly appealing because labeling data is often the most costly part, while acquiring samples  $\mathbf{x} \sim \mathcal{X}$  is easier. Unlabeled samples are thus valuable.

### 1.3.2 Statistics computation

#### General principle

Given the original feature matrix and the sampled knockoffs  $X, \tilde{X} \in \mathbb{R}^{n \times p}$ , statistics  $w_j$  for all  $j \in \{1, \dots, p\}$  are computed. Each  $w_j$  represents how more significant the original feature  $X_j$  is compared to  $\tilde{X}_j$ . These statistics must satisfy the *flip-sign* technical condition 3 for the FDR control to carry out, but a wide variety of choices is possible as will be shown.

**Definition 3.** (*flip-sign property*) A statistics function  $\omega: \mathbb{R}^{n \times 2p} \times \mathbb{R}^n \rightarrow \mathbb{R}^p$  is said to follow the *flip-sign* property if for any  $S \subseteq \{1, \dots, p\}$  and any  $j \in \{1, \dots, p\}$ ,

$$\omega_j([X, \tilde{X}]_{\text{swap}(S)}, \mathbf{y}) = \begin{cases} -\omega_j([X, \tilde{X}], \mathbf{y}) & \text{if } j \in S. \\ \omega_j([X, \tilde{X}], \mathbf{y}) & \text{otherwise.} \end{cases}$$

This property is very natural, as it is simply asking the original and the knockoff features to play antisymmetric roles.

#### Statistics aggregation

Constructing statistics satisfying the *flip-sign* property 3 is actually straightforward as an elementary scheme in two steps leads to such statistics. The idea is to build statistics for each original and each knockoff feature, and then aggregate them.

1. First construct statistics  $[z; \tilde{z}] = \zeta([X, \tilde{X}], \mathbf{y})$  with some function  $\zeta: \mathbb{R}^{n \times 2p} \times \mathbb{R}^n \rightarrow \mathbb{R}^{2p}$  satisfying

$$[z; \tilde{z}]_{\text{swap}(S)} = \zeta([X, \tilde{X}]_{\text{swap}(S)}, \mathbf{y}), \quad \forall S \subseteq \{1, \dots, p\} \quad (1.7)$$

The statistics  $z_j$  (resp.  $\tilde{z}_j$ ) indicates how significant the original (resp. knockoff) feature  $j$  is.

2. Then aggregate for each  $j \in \{1, \dots, p\}$  the statistics of the original feature  $z_j$  and the one of the corresponding knockoff  $\tilde{z}_j$  with an antisymmetric function  $a_j: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ . That is, set  $w_j = a_j(z_j, \tilde{z}_j)$ .

It is easy to show that such constructed statistics will satisfy the *flip-sign* property 3.

Basically any antisymmetric function  $a_j$  could work, but some choices experimentally lead to a better power. Here are a few examples of mappings:

- $w_j = z_j - \tilde{z}_j$  (experimentally gives highest power in many cases)
- $w_j = \max(z_j, \tilde{z}_j) \times \text{sign}(z_j - \tilde{z}_j)$  (first proposed in 2015)
- $w_j = \log \frac{z_j}{\tilde{z}_j}$

As for the function  $\zeta$ , it only needs to satisfy the *swap* property 1.7. This condition may seem restrictive but a large number of choices are actually valid.

#### Examples

A few examples of statistics relying on the aggregation trick are presented in this subsection.

In [14], Barber-Candès suggest after empirical observations the use of the Lasso Signed Max (LSM) statistics defined as follows.

$$z_j = \sup\{\lambda \mid \hat{\beta}_j(\lambda) \neq 0\}, \quad \hat{\beta}(\lambda) = \arg \min_{\mathbf{b}} \frac{1}{2} \|\mathbf{y} - [X, \tilde{X}]\mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_1 \quad (1.8)$$

The vector  $\hat{\beta}(\lambda) \in \mathbb{R}^{2p}$  contains the coefficients of a Lasso model with penalty coefficient  $\lambda > 0$ . As mentioned in subsection 1.1.3, all the coefficients are null starting from  $\lambda = +\infty$ , and are likely to shift (and to grow in absolute value) as  $\lambda \rightarrow 0$ . It makes sense to use the first point where the coefficient  $\beta_j$  is not null as a significance metric for each individual feature  $j$ . In addition, the LARS algorithms is able to compute that value pretty efficiently [19].

Training a plain Lasso estimator. Once again, the LARS algorithm allows to cross-validate the Lasso efficiently. Experimentally, the Lasso gives remarkable results in terms of power.

More generally, the coefficients in absolute value of any reasonable regressor (or classifier, depending on the task) constitute a judicious option of statistics. The fact that the *flip-sign* property isn't restrictive

makes the knockoff framework very powerful. Depending on the data, an adapted model can be trained be it logistic regression, SVMs, or random forests [7]. When the model has hyper-parameters, those are typically tuned using cross-validation. This step is usually costly.

### 1.3.3 Selection thresholds

The selection itself requires the computation of a data-dependent threshold  $\tau$  conditioned by the target FDR  $q \in [0, 1]$ . Finally, only the features  $j$  whose statistics  $w_j$  is above the threshold are selected.

$$\hat{\mathcal{S}} = \{j \mid w_j \geq \tau\} \quad (1.9)$$

Depending on how restrictive the procedure ought to be, the threshold  $\tau$  can be adapted. It leads to different guarantees regarding the control of the FDR. Barber-Candès establish two selection procedures, namely *knockoff* and *knockoff+*. They control the modified FDR (as defined below) and the FDR respectively.

**Definition 4.** Given an estimate  $\hat{\mathcal{S}}$  of  $\mathcal{S}$  and a desired target false discovery rate  $q \in [0, 1]$ , we define the modified FDR as follows:

$$mFDR = \mathbb{E} \frac{|\{j \mid j \in \hat{\mathcal{S}} \setminus \mathcal{S}\}|}{|\hat{\mathcal{S}}| + 1/q}$$

As  $0 \leq q \leq 1$ , mFDR as defined in 4 is always smaller than the actual FDR. It means that controlling the mFDR is less restrictive than controlling the FDR, as the FDR could potentially be arbitrarily larger than the mFDR. But if the target threshold  $q$  is not too small, and if many features are selected by the procedure, the modified version of the FDR is close enough to the actual FDR. As will be shown later, being less restrictive by controlling only the mFDR can greatly improve the power of the selection.

The only difference between the *knockoff* and the *knockoff+* procedures is the selection threshold  $\tau$ .

**Definition 5.** (*knockoff and knockoff+ thresholds*) Given the statistics  $\mathbf{w} \in \mathbb{R}^p$  computed from  $X$  and  $\tilde{X}$ , we define the *knockoff* and *knockoff+* threshold respectively as follows:

$$\tau = \min \left\{ t > 0 \mid \frac{|\{j \mid w_j \leq -t\}|}{|\{j \mid w_j \geq t\}|} \leq q \right\} \quad (1.10)$$

$$\tau^+ = \min \left\{ t > 0 \mid \frac{1 + |\{j \mid w_j \leq -t\}|}{|\{j \mid w_j \geq t\}|} \leq q \right\} \quad (1.11)$$

They only differ by their numerator, where  $\tau^+$  has an additional +1. The main result of Barber-Candès regarding the control of the mFDR and FDR is stated in 3.

**Theorem 3.** (*guarantees of the knockoff procedures*) Construct  $\hat{\mathcal{S}} = \{j \mid w_j \geq \tau\}$  and  $\hat{\mathcal{S}}^+ = \{j \mid w_j \geq \tau^+\}$ . These selections ensure the following FDR controls:

$$mFDR[\hat{\mathcal{S}}] \leq q, \quad FDR[\hat{\mathcal{S}}^+] \leq q \quad (1.12)$$

### 1.3.4 Bottlenecks

Despite the nice theoretical guarantees on the FDR control that the knockoff procedure proposes, two bottlenecks hurt its performances in the high dimensional setting.

#### SDP

In the gaussian setting, knockoff features are sampled according to the distribution 1.6. The control guarantees hold for any  $\mathbf{s} \in \mathbb{R}^p$  such that the covariance matrix  $\Omega$  is semidefinite positive.

**Proposition 4.** Let  $\Omega = \begin{bmatrix} \Sigma & \Sigma - \text{diag } \mathbf{s} \\ \Sigma - \text{diag } \mathbf{s} & \Sigma \end{bmatrix}$ . Then  $\Omega \succeq \mathbf{0}_{p \times p}$  if and only if  $2\Sigma \succeq \text{diag } \mathbf{s} \succeq \mathbf{0}$ .

*Proof.* Note  $D = \text{diag } \mathbf{s}$ . By computing the Schur complement [32]  $\Omega_{/\Sigma} = 2D - D\Sigma^{-1}D$ ,  $\Omega \succeq \mathbf{0}$  is equivalent to  $\Omega_{/\Sigma} \succeq \mathbf{0}$ . Define  $M$  and its Schur complements as follows

$$M = \begin{bmatrix} 2D & D \\ D & \Sigma \end{bmatrix}, \quad \begin{cases} M_{/2D} = \Sigma - \frac{1}{2}D \\ M_{/\Sigma} = 2D - D\Sigma^{-1}D \end{cases}$$

Finally,  $\Omega_{/\Sigma} = M_{/\Sigma}$  is p.s.d. if and only if both  $\Sigma - \frac{1}{2}D$  and  $D$  are p.s.d.  $\square$

## Statistics computation

and could take advantage of sparse naive Bayes to run faster.

These two bottlenecks make the . The following chapters tackle these two issues by proposing fast statistics computation. From now, only binary classification problems will be considered, where the label  $y$  takes values in  $\{\mathcal{C}^+, \mathcal{C}^-\}$ .

### 1.3.5 Python implementation

Barber-Candès (along with other coauthor) provided a R implementation of the knockoff framework<sup>1</sup>. To the best of our knowledge, no unified Python implementation Scikit-Learn [24]

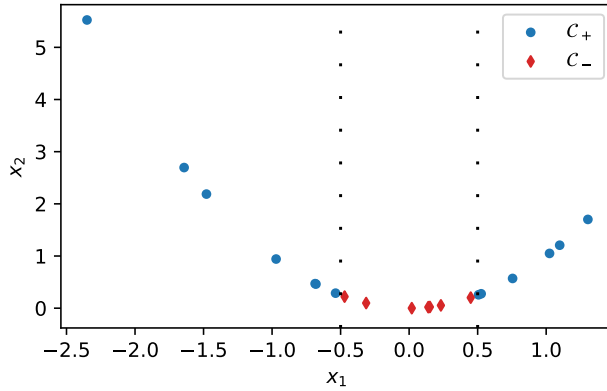


FIGURE 1.1: Illustration of the non-uniqueness of the Markov blanket. Here  $\mathcal{X}_1 \sim \mathcal{N}(0, 1)$ ,  $\mathcal{X}_2 = \mathcal{X}_1^2$ , and  $\mathcal{Y}$  is  $\mathcal{C}_+$  if  $|\mathcal{X}_1| \geq \frac{1}{2}$ ,  $\mathcal{C}_-$  otherwise. Both  $\mathcal{X}_1$  and  $\mathcal{X}_2$  alone are enough to predict  $\mathcal{Y}$ .

<sup>1</sup> The package page can be found at this address <https://cran.r-project.org/web/packages/knockoff/index.html>.

## Chapter 2

# Sparse naive Bayes

Naive Bayes was first introduced in the early 60s by [20] for text documents classification. Despite its naive assumptions (independence of the features), naive Bayes remains an engaging model for large scale datasets because of its low complexity. The time complexity to train a model is  $\mathcal{O}(n \cdot p)$ , where  $n$  and  $p$  are the number of samples and the number of features respectively. Another appealing propriety is that naive Bayes can be trained in an online fashion, as new data points come in a sequential order. It can be particularly helpful if the dataset doesn't fit in memory. Furthermore, distributed implementations could even be considered in order to speed up the learning process.

In this chapter, classical naive Bayes along with a few notations are briefly introduced. Then, a sparse version of naive Bayes allowing in particular to perform feature selection is presented.

## 2.1 Reminders on vanilla naive Bayes

In this section, we recall briefly the naive Bayes model in the specific case of binary classification, and its training under the bernoulli, multinomial and gaussian underlying assumptions. Most key results can naturally be extended to the multiple classes setting.

Let  $n$  and  $p$  be two integers,  $X \in \mathbb{R}^{n \times p}$  a feature matrix and  $\mathbf{y} \in \{-1, 1\}^n$  the associated target vector. The negative and the positive classes are noted  $\mathcal{C}_-$  and  $\mathcal{C}_+$  respectively, and will be indifferently substituted with their respective labels.

### 2.1.1 General settings

Note  $(\mathcal{X}, \mathcal{Y})$  the pair of random variables that generated the samples and targets  $X$  and  $\mathbf{y}$ . The goal is to explain  $\mathbf{y}$  given  $X$ , that is, finding the posterior probabilities  $\Pr(\mathcal{C}_\pm | \mathcal{X})$ . Given these probabilities, a new observation  $\mathbf{x} \in \mathbb{R}^p$  is classified according to the highest posterior probability between the two classes

$$y(\mathbf{x}) = \arg \max_{c \in \{\mathcal{C}_-, \mathcal{C}_+\}} \Pr(\mathcal{C} | \mathbf{x}) \quad (2.1)$$

To do so, we combine the use of Bayes rule and make the (very) naive assumption that all the features are independent given the class, that is

$$\Pr(\mathcal{X} = \mathbf{x} | \mathcal{C}_\pm) = \prod_{j=1}^p \Pr(\mathcal{X}_j = x_j | \mathcal{C}_\pm), \quad \Pr(\mathcal{C}_\pm | \mathcal{X} = \mathbf{x}) = \frac{\Pr(\mathcal{X} = \mathbf{x} | \mathcal{C}_\pm) \cdot \Pr(\mathcal{C}_\pm)}{\Pr(\mathcal{X} = \mathbf{x})} \quad (2.2)$$

On the right hand side, the denominator  $\Pr(\mathcal{X} = \mathbf{x})$  does not depend on the class  $\mathcal{C}_\pm$ . It is therefore not required to evaluate it in order to perform the inference described in 2.1. Thus, we only need to estimate the probabilities  $\Pr(\mathcal{C}_\pm)$  and  $\Pr(\mathbf{x} | \mathcal{C}_\pm)$ . The former are simply data averages, or the frequency of the positive and negative classes in the observed data. As for the latter probabilities  $\Pr(\mathbf{x} | \mathcal{C}_\pm) = \prod_{j=1}^p \Pr(x_j | \mathcal{C}_\pm)$ , they can be modeled by a plethora of distribution families, depending on the prior knowledge we have on the data. The distribution is typically parametrized by some vector  $\boldsymbol{\theta} \in \mathbb{R}^m$ . The probabilities are usually computed by maximizing the likelihood  $\mathcal{L}$ , or equivalently the log-likelihood  $\mathcal{LL} = \log \mathcal{L}$ , of the observed data.

$$\mathcal{LL}(\boldsymbol{\theta}) = \sum_{i=1}^n \Pr(\mathbf{x}_i | y_i; \boldsymbol{\theta})$$

We present now 3 meaningful cases, where the prior distributions of  $\Pr(\mathbf{x} | \mathcal{C}_\pm)$  are either gaussian, bernoulli or multinomial. Let  $\mathcal{I}_\pm = \{i \in \{1, \dots, n\} | y_i = \mathcal{C}_\pm\}$  be the sets containing the indexes of the positive and negative data points. We also note for each class  $\mathcal{C}_\pm$  their cardinalities and empirical sums respectively, as follows:

$$n^\pm = |\mathcal{I}_\pm|, \quad \mathbf{f}^\pm = \sum_{i \in \mathcal{I}_\pm} \mathbf{x}_i,$$

### 2.1.2 Gaussian naive Bayes

In this case the observed data, conditioned on its label, is modeled by the gaussian distribution  $\mathcal{N}(\boldsymbol{\mu}^\pm, \Sigma^\pm)$ . It is the most common case, to account for continuous data. Note that the covariance  $\Sigma^\pm \in \mathbb{R}^{p \times p}$  is diagonal because of the independence assumption that we made.

$$\Pr(\mathbf{x} \mid C_\pm) = \frac{1}{\sqrt{(2\pi)^p \det(\Sigma_\pm)}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_\pm)^\top \Sigma_\pm^{-1}(\mathbf{x} - \boldsymbol{\mu}_\pm)\right)$$

By denoting  $\sigma_j = \Sigma_{jj}$ , the log-likelihood can be written

$$\begin{aligned} \mathcal{LL}_g(\boldsymbol{\mu}_+, \boldsymbol{\sigma}_+, \boldsymbol{\mu}_-, \boldsymbol{\sigma}_-) &= \sum_{j=1}^p \left[ \sum_{i \in \mathcal{I}_+} -\frac{1}{2} \log(2\pi) - \log \sigma_j^+ - \frac{(x_j - \mu_j^+)^2}{2\sigma_j^{+2}} \right. \\ &\quad \left. + \sum_{i \in \mathcal{I}_-} -\frac{1}{2} \log(2\pi) - \log \sigma_j^- - \frac{(x_j - \mu_j^-)^2}{2\sigma_j^{-2}} \right] \end{aligned}$$

The minimizer of  $\mathcal{LL}_g$  admits a closed-form solution:

$$\boldsymbol{\mu}^\pm = \frac{\mathbf{f}^\pm}{n^\pm}, \quad \boldsymbol{\sigma}^\pm = \sqrt{\frac{1}{n^\pm} \sum_{i \in \mathcal{I}^\pm} (\mathbf{x}_i - \boldsymbol{\mu}^\pm)^2}$$

### 2.1.3 Bernoulli naive Bayes

The Bernoulli distribution assumes that the design matrix is binary, that is  $X \in \{0, 1\}^{n \times p}$ . Even though this assumption case isn't very useful in practice, its solution is simple and elegant. Assume In order to model the conditional probabilities, we may assume the existence of  $\boldsymbol{\theta}^+, \boldsymbol{\theta}^- \in (0, 1)^p$  such that for any data point  $\mathbf{x} \in \mathbb{R}^p$ ,

$$\Pr(x_j \mid C_\pm) = (\theta_j^\pm)^{x_j} \cdot (1 - \theta_j^\pm)^{1-x_j}$$

It yields that  $\log \Pr(\mathbf{x} \mid C_\pm) = \mathbf{x}^\top \log \boldsymbol{\theta}^\pm + (\mathbf{1} - \mathbf{x})^\top \log(\mathbf{1} - \boldsymbol{\theta}^\pm)$  and finally

$$\begin{aligned} \mathcal{LL}_b(\boldsymbol{\theta}^+, \boldsymbol{\theta}^-) &= \sum_{i \in \mathcal{I}^+} \log \Pr(\mathbf{x}_i \mid C_+) + \sum_{i \in \mathcal{I}^-} \log \Pr(\mathbf{x}_i \mid C_-) \\ &= \mathbf{f}_+^\top \log \boldsymbol{\theta}^+ + (n_+ \mathbf{1} - \mathbf{f}_+)^\top \log(\mathbf{1} - \boldsymbol{\theta}^+) \\ &\quad + \mathbf{f}_-^\top \log \boldsymbol{\theta}^- + (n_- \mathbf{1} - \mathbf{f}_-)^\top \log(\mathbf{1} - \boldsymbol{\theta}^-) \end{aligned} \quad (2.3)$$

The independence assumption makes the optimization problem decomposable across features; it reduces to  $p$  simpler maximizations, each of them admitting a closed-form solution. Finally, we find that

$$\theta_\pm^* = \frac{f_\pm^\pm}{n^\pm}, \quad \text{which is simply the average of each class.}$$

### 2.1.4 Multinomial naive Bayes

Multinomial naive Bayes is more general than the Bernoulli version as we suppose that  $X \in \mathbb{N}^{n \times p}$  is generated by the following underlying distribution

$$\Pr(\mathbf{x} \mid C_\pm) = \frac{(\sum_{j=1}^p x_j)!}{\prod_{j=1}^p x_j!} \cdot \prod_{j=1}^p \theta_\pm^{x_j}$$

It is parametrized by  $\boldsymbol{\theta}^+, \boldsymbol{\theta}^- \in (0, 1)^p$ , and they must satisfy  $\mathbf{1}^\top \boldsymbol{\theta}^+ = \mathbf{1}^\top \boldsymbol{\theta}^- = 1$  for it to be a proper distribution. Note that this model is still valid in the more general case where we only assume the data to be non-negative,  $X \in \mathbb{R}_+^{n \times p}$ . Hence, it is not as restrictive as it may appear at first sight, as it is applicable to a large number of datasets. The log probability is given by

$$\log \Pr(\mathbf{x} \mid C_\pm) = \mathbf{x}^\top \log \boldsymbol{\theta}_\pm + \log \frac{(\sum_{j=1}^p x_j)!}{\prod_{j=1}^p x_j!}$$

and the log-likelihood reduces to

$$\mathcal{LL}_m(\boldsymbol{\theta}^+, \boldsymbol{\theta}^-) = \mathbf{f}_+^\top \log \boldsymbol{\theta}^+ + \mathbf{f}_-^\top \log \boldsymbol{\theta}^- \quad (2.4)$$

which is again decomposable across features. It turns out that  $\boldsymbol{\theta}_\pm^* = \frac{\mathbf{f}_\pm^\pm}{\mathbf{1}^\top \mathbf{f}_\pm^\pm}$ .



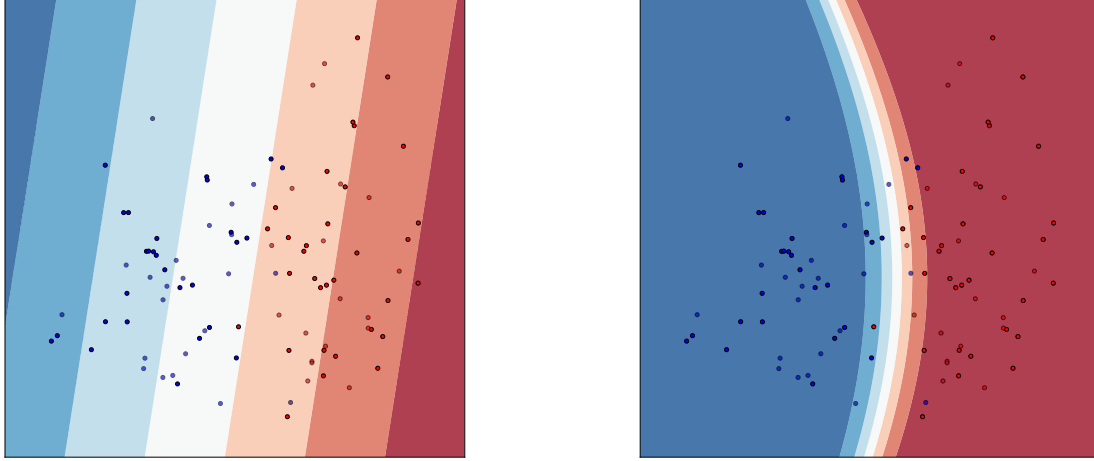


FIGURE 2.1: Illustration of the decision boundary for the logistic regression (left) and gaussian naive Bayes (right) for binary classification. For logistic regression, the separation is an hyperplane. For gaussian naive Bayes, the separation has a dependency on  $x \odot x$  and can be non-linear. Both bernoulli and multinomial naive Bayes would have a linear separation.

In the models presented above, the time complexity to train the naive Bayes classifier is  $\mathcal{O}(n \cdot p)$ . With a larger number of classes besides  $\mathcal{C}^-$  and  $\mathcal{C}^+$ , say  $k$  classes, this complexity is multiplied by  $k$ . Besides this low asymptotic complexity, the solutions can be computed in closed-form, which makes the effective computation cost very low. In comparison, no closed-form solution exist for the Lasso, for logistic regression [25], or for SVMs [9]. For these models, a more costly gradient-descent based algorithm is executed to find the global optimum.

### 2.1.5 Decision boundary

Given a new data point  $\mathbf{x} \in \mathbb{R}^p$ , we wish to attach it the most probable label  $y \in \{\mathcal{C}^-, \mathcal{C}^+\}$ . No matter what model parametrized by  $\boldsymbol{\theta}$  was chosen, 2.1 reduces to

$$\begin{aligned} y &= \arg \max_{c \in \{\mathcal{C}^-, \mathcal{C}^+\}} \Pr(\mathcal{C} | \mathbf{x}) \\ &= \text{sign} \log \frac{\Pr(\mathcal{C}^+ | \mathbf{x}; \boldsymbol{\theta})}{\Pr(\mathcal{C}^- | \mathbf{x}; \boldsymbol{\theta})} \\ &= \text{sign} \left[ \log \frac{\Pr(\mathcal{C}^+)}{\Pr(\mathcal{C}^-)} + \log \frac{\Pr(\mathbf{x} | \mathcal{C}^+)}{\Pr(\mathbf{x} | \mathcal{C}^-)} \right] \end{aligned}$$

In the cases of *bernoulli* (2.1.3) and *multinomial* (2.1.4) naive Bayes, there exist  $v \in \mathbb{R}$  and  $\mathbf{w} \in \mathbb{R}^p$  such that  $y = \text{sign}(v + \mathbf{w}^\top \mathbf{x})$ . In these two special cases, the decision boundary is a hyperplane (which doesn't happen for gaussian naive Bayes). For both of them  $v$  has the same value, and by noting  $\mathbf{w}_b$  and  $\mathbf{w}_m$  the weights for the bernoulli and the multinomial case respectively, we have

$$v = \log \frac{\Pr(\mathcal{C}^+)}{\Pr(\mathcal{C}^-)}, \quad \begin{cases} \mathbf{w}_b = \log(\boldsymbol{\theta}^+ \odot (\mathbf{1} - \boldsymbol{\theta}^-)) - \log(\boldsymbol{\theta}^- \odot (\mathbf{1} - \boldsymbol{\theta}^+)) \\ \mathbf{w}_m = \log \boldsymbol{\theta}^+ - \log \boldsymbol{\theta}^- \end{cases}$$

Figure 2.1 compares the decision boundary of the logistic regression to the one of the gaussian naive Bayes.

## 2.2 Sparse naive Bayes

A sparse version of naive Bayes was introduced in 2019 [3]. They add a sparsity constraint in the Bernoulli and the multinomial optimization problems portrayed above. It imposes the weight vector  $\mathbf{w}$  to have a number of non-zero entries under a certain threshold  $k \in \mathbb{N}$ . That property is similar to the one of the Lasso presented in 1.1.3. But the sparsity in sparse naive Bayes (SNB) is controlled by an integer  $k$ , the exact sparsity level desired on the weight vector, while the Lasso relies on a penalty coefficient  $\lambda \in \mathbb{R}$ . This sparsity property makes SNB employable to perform feature selection, by keeping only the features whose weight is non-zero. Next sections detail the problem statement, the main results of the authors, and some applications.

### 2.2.1 Problem statement

Let  $0 \leq k \leq p$  be a desired level of sparsity. We wish to train a naive Bayes classifier whose decision boundary depends on at most  $k$  features. For any  $\mathbf{v} \in \mathbb{R}^n$  we note  $\|\mathbf{v}\|_0$  the number of non-zero entries (the cardinality) of the vector. As shown in the previous section, for both bernoulli and multinomial naive Bayes, there exist  $v \in \mathbb{R}$  and  $\mathbf{w} \in \mathbb{R}^p$  such that the prediction  $y(\mathbf{x})$  for a new data point  $\mathbf{x} \in \mathbb{R}^p$  is  $\text{sign}(v + \mathbf{w}^\top \mathbf{x})$ . Furthermore, the  $j$ th entry of the decision vector  $\mathbf{w}$  is null if and only if  $\theta_j^+ = \theta_j^-$ , where  $\theta^+$  and  $\theta^-$  are the parameters of the loss functions  $\mathcal{LL}_b$  and  $\mathcal{LL}_m$  in 2.3 and 2.4 respectively. Naturally, imposing the constraint  $\|\theta^+ - \theta^-\|_0 \leq k$  in the optimization problems will yield a weight vector  $\mathbf{w}$  with the desired sparsity. The optimization problems for Bernoulli and multinomial SNB can be phrased as follows:

$$\begin{aligned} & \underset{\theta^+, \theta^-}{\text{maximize}} && \mathcal{LL}_b(\theta^+, \theta^-) && \underset{\theta^+, \theta^-}{\text{maximize}} && \mathcal{LL}_m(\theta^+, \theta^-) \\ & \text{subject to} && \|\theta^+ - \theta^-\|_0 \leq k. && \text{subject to} && \|\theta^+ - \theta^-\|_0 \leq k \\ & && && \text{and} && \mathbf{1}^\top \theta^+ = \mathbf{1}^\top \theta^- = 1. \end{aligned} \quad (\text{abc})$$

### 2.2.2 Main results and resolution

Surprisingly, despite the combinatorial constraints, this optimization problem can be (approximately) solved very efficiently, with an additional minor cost compared to vanilla naive Bayes. For the Bernoulli case especially, an optimal solution can be computed in closed-form as shown in Theorem 5.

**Theorem 5.** *Supposing that  $X \in \{0, 1\}^{n \times p}$  is modeled by the Bernoulli distribution. Then exact solution to the problem abc can be computed. First, define  $\mathbf{v}$  and  $\mathbf{w}$  as follows*

$$\begin{aligned} \mathbf{t} &= (\mathbf{f}^+ + \mathbf{f}^-) \odot \log\left(\frac{\mathbf{f}^+ + \mathbf{f}^-}{n}\right) + (n\mathbf{1} - \mathbf{f}^+ - \mathbf{f}^-) \odot \log\left(\mathbf{1} - \frac{\mathbf{f}^+ + \mathbf{f}^-}{n}\right) \\ \mathbf{u} &= \mathbf{f}^+ \odot \log \frac{\mathbf{f}^+}{n^+} + (n^+ \mathbf{1} - \mathbf{f}^+) \odot \log\left(\mathbf{1} - \frac{\mathbf{f}^+}{n^+}\right) \\ &\quad + \mathbf{f}^- \odot \log \frac{\mathbf{f}^-}{n^-} + (n^- \mathbf{1} - \mathbf{f}^-) \odot \log\left(\mathbf{1} - \frac{\mathbf{f}^-}{n^-}\right) \end{aligned}$$

Let  $\mathcal{I}$  be the set of  $p - k$  smallest elements of  $\mathbf{u} - \mathbf{t}$ , and let

$$\theta_{\star j}^+ = \theta_{\star j}^- = \frac{1}{n}(f_j^+ + f_j^-) \quad \forall j \in \mathcal{I}, \quad \theta_{\star j}^\pm = \frac{f_j^\pm}{n^\pm} \quad \forall j \notin \mathcal{I}$$

Forming the vectors  $\mathbf{f}^-$  and  $\mathbf{f}^+$  is very quick and takes asymptotically  $\mathcal{O}(n \cdot p)$ . The constructing  $\mathbf{t}$  and  $\mathbf{u}$  can be done in  $\mathcal{O}(p)$  steps. Finding the  $k$  largest elements of  $\mathbf{u} - \mathbf{t}$  takes  $\mathcal{O}(p \cdot \log k)$  steps. Finally, constituting  $\theta_\pm^*$  requires  $\mathcal{O}(p)$  steps. In total, the maximizer can be found in  $\mathcal{O}(n \cdot p + p \cdot \log k)$  steps, which is very close to the cost  $\mathcal{O}(n \cdot p)$  of naive Bayes.

In the multinomial case, there is no closed-form solution, but a near-optimal one can be obtained as stated in Theorem 6.

**Theorem 6.** *Suppose that  $X \in \mathbb{R}_+^{n \times p}$  is modeled by the multinomial distribution. Define  $\phi_k : \alpha \mapsto s_k(\mathbf{h}(\alpha)) + C$  where  $C$  is some constant,  $s_k$  the sum of the  $k$  largest values of a vector, and*

$$\begin{aligned} \mathbf{h}(\alpha) &= \mathbf{f}_+ \odot \log \mathbf{f}_+ + \mathbf{f}_- \odot \log \mathbf{f}_- - (\mathbf{f}_+ + \mathbf{f}_-) \odot \log(\mathbf{f}_+ + \mathbf{f}_-) \\ &\quad - \mathbf{f}_+ \log \alpha - \mathbf{f}_- \log(1 - \alpha) \end{aligned}$$

Let  $\alpha^*$  be the minimizer of  $\phi_k$ ,  $\mathcal{I}$  the set of the  $p - k$  smallest entries of  $\mathbf{h}(\alpha^*)$ , and  $B_\pm = \sum_{j \notin \mathcal{I}} f_j^\pm$ . A primal point can be reconstructed as follows:

$$\theta_{\star j}^+ = \theta_{\star j}^- = \frac{f_j^+ + f_j^-}{\mathbf{1}^\top(\mathbf{f}^+ + \mathbf{f}^-)} \quad \forall j \in \mathcal{I}, \quad \theta_{\star j}^\pm = \frac{B_+ + B_-}{B_\pm} \frac{f_j^\pm}{\mathbf{1}^\top(\mathbf{f}^+ + \mathbf{f}^-)} \quad \forall j \notin \mathcal{I}$$

Furthermore, it holds that  $\psi(k - 4) \leq \phi(k) \leq \psi(k) \leq \phi(k + 4)$ , implying that the duality gap is small if  $\psi(k) - \psi(k - 4)$  is small.

Experimentally, the duality gap quickly converges to 0 as  $k$  increases, and the reconstructed primal point is near-optimal. The time complexity is once again  $\mathcal{O}(n \cdot p + p \cdot \log k)$ , which is a minor additional cost compared to plain naive Bayes.

The authors experiment SNB on several text datasets, including AMZN, IMDB, TWTR, MPQA and SST2. They compare it with more costly methods like the Lasso,  $\ell_1$ -penalized logistic regression and SVMs. They obtain competitive test accuracies, while training their models several order of magnitude faster.

Samples	Total features	Numerical features	Categorical features	Features after encoding
45 000 000	39	13	26	10 000 000

TABLE 2.1: Criteo dataset characteristics. Even though the number of features is small, most categorical features have dozens of thousands of categories. It makes the training of predicting models particularly challenging as it requires several dozens of GB of memory.

## 2.3 Applications

The apparent low complexity of sparse naive Bayes compared to  $\ell_1$ -penalized methods such as Lasso, logistic regression or SVMs makes is appealing for very large scale datasets. We mention here a few applications to which we will come back later.

### 2.3.1 Criteo dataset

As part of a Kaggle competition<sup>1 2</sup> hosted by CriteoLabs in mid-2014 in Table 2.1

---

<sup>1</sup> Link to the Kaggle competition: <https://www.kaggle.com/c/criteo-display-ad-challenge>.

<sup>2</sup> The competition's dataset can be downloaded at this address: <https://labs.criteo.com/2014/02/download-kaggle-display-advertising-challenge-dataset/>.



## Chapter 3

# Fast statistics computation

The computation of statistics is a key step in the knockoff selection process described in section 1.3. As it has been discussed, a large variety of statistics control the FDR, provided that the *flip-sign* property 3 is satisfied.

However, the power can drastically change depending on how good the statistics are. We also suppose that the knockoffs were generated respecting the conditions. . . . Experimentally, it appears that for most tasks the coefficients of a cross-validated Lasso

In this chapter, we note  $X \in \mathbb{R}^{n \times p}$  the original design matrix, and we suppose that the distribution  $\tilde{\mathcal{X}} \mid \mathcal{X}$  could be computed. Let  $\tilde{X} \in \mathbb{R}^{n \times p}$  be sampled from that knockoff distribution.

### 3.1 Lasso is good but slow

### 3.2 Fast statistics

### 3.3 Multi-stage procedures

Multi-stage procedure [30]



## Chapter 4

# SDP

In the case of gaussian knockoffs, or more generally when we only impose  $[X, \tilde{X}]_{\text{swap}(S)}$  to have the same first two moments as  $[X, \tilde{X}]$  for any subset  $S \subseteq \{1, \dots, p\}$ , rather than the same distribution [S.1](#),  $\tilde{X}$  can be sampled from a normal distribution  $\mathcal{N}(\mathbf{v}, \Upsilon)$  whose parameters  $\mathbf{v}, \Upsilon$  are formulated in [1.6](#). These formulas are valid for any vector  $\mathbf{s} \in \mathbb{R}^p$  such that  $\Omega$  is indeed a covariance matrix (semidefinite positive).

$$\Omega = \begin{bmatrix} \Sigma & \Sigma - \text{diag } \mathbf{s} \\ \Sigma - \text{diag } \mathbf{s} & \Sigma \end{bmatrix} \succeq \mathbf{0}_{2p \times 2p}$$

As shown in proposition [4](#), this matrix is semidefinite positive if and only if  $\mathbf{0}_{p \times p} \preceq \text{diag } \mathbf{s} \preceq 2\Sigma$ . The first inequality clearly holds if all the entries of  $\mathbf{s}$  are positive. The second one is trickier.

In this chapter, we show why the choice of  $\mathbf{s}$  is important and how to efficiently find a good one in the high dimensional setting. In the remaining, we suppose that we know  $\Sigma$ , even though in practice the covariance would be estimated  $\hat{\Sigma}$ .

### 4.1 Equi-correlated knockoffs

#### A cheap solution

For any psd matrix  $A \in \mathbb{R}^p$ ,  $A + \alpha \cdot \mathbf{I}$  has the same eigenvalues as  $A$ , but shifted by  $\alpha$ . It gives a fast and simple way to find a suitable  $\mathbf{s}$ .

$$s_j = \min(2\lambda_{\min}(\Sigma), 1) \quad \text{for all } 0 \leq j \leq p \quad (4.1)$$

The problem of finding the smallest eigenvalue  $\lambda_{\min}(\Sigma)$  can be solved efficiently. Example.

#### Why it is not desirable

For large values of  $p$ , the minimum eigenvalue of  $\Sigma$  is very likely to be small, unless  $\Sigma$  has a substantially special structure. Let us analyze briefly the covariance of  $[\mathcal{X}; \tilde{\mathcal{X}}]$  to understand why it is not desirable.

$$\begin{cases} \text{cov}(\mathcal{X}_j, \tilde{\mathcal{X}}_{j'}) = \text{cov}(\mathcal{X}_j, \mathcal{X}_{j'}) \text{ for all } j \neq j' \\ \text{cov}(\mathcal{X}_j, \tilde{\mathcal{X}}_j) = \text{cov}(\mathcal{X}_j, \mathcal{X}_j) - s_j \text{ for all } j \end{cases}$$

First,  $\tilde{\mathcal{X}}$  has the same internal covariance as  $\mathcal{X}$ , and two distinct original and knockoff features have the same covariance as the one of the two original ones. It makes the knockoff feature sufficiently close to the original features to fool them. However, an original feature  $j$  and its corresponding knockoff that is smaller the larger  $s_j$  is. If  $s_j$  was 0,  $\mathcal{X}_j$  couldn't be distinguished from  $\tilde{\mathcal{X}}_j$  and the selection would suffer from a low power.

### 4.2 SDP knockoffs

This observation motivates us to maximize the entries of  $\mathbf{s}$ , while maintaining the inequality  $\text{diag } \mathbf{s} \preceq 2\Sigma$ . This can be formulated in the optimization problem [4.2](#).

$$\arg \max_{\mathbf{s} \in \mathbb{R}^p} \sum_{j=1}^p s_j \quad \text{subject to} \quad \begin{cases} s_j \geq 0 \text{ for all } j \\ \text{diag } \mathbf{s} \preceq 2\Sigma \end{cases} \quad (4.2)$$

This problem is a structured semidefinite program (SDP) and can efficiently be solved for small values of  $p$  by interior point methods [\[21\]](#) for example. For larger values of  $p$ , even first order methods like SCS [\[22\]](#)

or alternating direction [31] quickly become intractable. Even though the convergence speed depends a lot on  $\Sigma$ , it experimentally appears that alternative methods have to be considered when  $p > 1000$ .

In order to reduce the computation time, Barber-Candès suggest to solve an approximated problem of 4.2 in 2 steps that we describe below.

**Step 1.** Pick an approximation  $\Sigma_{\text{approx}}$  of  $\Sigma$  and solve

$$\arg \max_{\hat{\mathbf{s}} \in \mathbb{R}^p} \sum_{j=1}^p \hat{s}_j \quad \text{subject to} \quad \begin{cases} \hat{s}_j \geq 0 \text{ for all } j \\ \text{diag } \hat{\mathbf{s}} \preceq 2\Sigma_{\text{approx}} \end{cases} \quad (4.3)$$

**Step 2.** Solve the one dimensional maximization

$$\arg \max_{\gamma \in \mathbb{R}} \gamma \quad \text{subject to} \quad \text{diag}(\gamma \cdot \hat{\mathbf{s}}) \preceq 2\Sigma \quad (4.4)$$

Finally, pick  $\mathbf{s} = \gamma \cdot \hat{\mathbf{s}}$ .

**Remark 7.** Note that the two extreme options  $\Sigma_{\text{approx}} = \mathbf{I}$  and  $\Sigma_{\text{approx}} = \Sigma$  yield the same solutions as solving equi-knockoffs 4.1 and full SDP knockoffs 4.2 respectively.

Solving 4.4 in step 2 can be done very efficiently using bisection as it is a one-dimensional SDP. The optimization problem 4.3 is however the same as the one in 4.2, except for the approximation  $\Sigma_{\text{approx}}$ . To speed up the computations,  $\Sigma_{\text{approx}}$  can be chosen to be a block-diagonal matrix of  $k$  blocks. The maximization then reduces to  $k$  smaller SDPs for which the solutions can be found more efficiently. These could even be distributed in several computation nodes. Picking  $\Sigma_{\text{approx}}$  is a compromise between available computation time and eventual power of the procedure. There is a priori no ideal way to do it.



## Appendix A

# An appendix

An appendix



# Bibliography

- [1] Constantin Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11:171–234, 01 2010.
- [2] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *American Statistician*, 46(3):175–185, 8 1992.
- [3] Armin Askari, Alexandre d’Aspremont, and Laurent El Ghaoui. Naive Feature Selection: Sparsity in Naive Bayes. *arXiv e-prints*, page arXiv:1905.09884, May 2019.
- [4] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate - a practical and powerful approach to multiple testing. *J. Royal Statist. Soc., Series B*, 57:289 – 300, 11 1995.
- [5] Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Ann. Stat.*, 29, 08 2001.
- [6] Mairead Bermingham, Ricardo Pong-Wong, Athina Spiliopoulou, Caroline Hayward, Igor Rudan, H. Campbell, A. Wright, J. Wilson, Felix Agakov, Pau Navarro, and Chris Haley. Application of high-dimensional feature selection: Evaluation for genomic prediction in man. *Scientific Reports*, 5:10312, 05 2015.
- [7] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.
- [8] Giuseppe C. Calafiore and Giulia Fracastoro. Sparse  $\ell_1$  and  $\ell_2$  Center Classifiers. page arXiv:1911.07320, Nov 2019.
- [9] Corinna Cortes and V. VAPNIK. Support-vector networks. *Chem. Biol. Drug Des.*, 297:273–297, 01 2009.
- [10] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(1):131 – 156, 1997.
- [11] Morris L. Eaton. *Multivariate Statistics: a Vector Space Approach*. 1983.
- [12] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least Angle Regression. *arXiv Mathematics e-prints*, page math/0406456, Jun 2004.
- [13] George Forman. An extensive empirical study of feature selection metrics for text classification [j]. *Journal of Machine Learning Research - JMLR*, 3, 03 2003.
- [14] Rina Foygel Barber and Emmanuel J. Candès. Controlling the false discovery rate via knockoffs. *arXiv e-prints*, page arXiv:1404.5609, Apr 2014.
- [15] Isabelle Guyon and André Elisseeff. An introduction of variable and feature selection. *J. Machine Learning Research Special Issue on Variable and Feature Selection*, 3:1157 – 1182, 01 2003.
- [16] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 01 2002.
- [17] Eui-Hong (Sam) Han and George Karypis. Centroid-based document classification: Analysis and experimental results. In Djamel A. Zighed, Jan Komorowski, and Jan Żytkow, editors, *Principles of Data Mining and Knowledge Discovery*, pages 424–431, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.
- [18] Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365 – 411, 2004.
- [19] Julien Mairal and B. Yu. Complexity analysis of the lasso regularization path. 1, 04 2012.
- [20] M. E. Maron. Automatic indexing: An experimental inquiry. *J. ACM*, 8:404–417, 1961.
- [21] Diskrete Optimierung, Christoph Helmberg, Franz Rendl, Robert Vanderbei, and Henry Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6, 02 1970.
- [22] Brendan O’donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *J. Optim. Theory Appl.*, 169(3):1042–1068, June 2016.

- [23] Judea Pearl. Chapter 3 - markov and bayesian networks: Two graphical representations of probabilistic knowledge. In Judea Pearl, editor, *Probabilistic Reasoning in Intelligent Systems*, pages 77 – 141. Morgan Kaufmann, San Francisco (CA), 1988.
- [24] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Edouard Duchesnay, and Gilles Louppe. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 01 2012.
- [25] Joanne Peng, Kuk Lee, and Gary Ingersoll. An introduction to logistic regression analysis and reporting. *Journal of Educational Research - J EDUC RES*, 96:3–14, 09 2002.
- [26] Jaime Roquero Gimenez and James Zou. Improving the Stability of the Knockoff Procedure: Multiple Simultaneous Knockoffs and Entropy Maximization. *arXiv e-prints*, page arXiv:1810.11378, Oct 2018.
- [27] Jiliang Tang, Salem Alelyani, and Huan Liu. *Feature selection for classification: A review*, pages 37–64. CRC Press, 1 2014.
- [28] Robert Tibshirani. Regression shrinkage and selection via the lasso. 1996.
- [29] Gerard V. Trunk. A problem of dimensionality: A simple example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:306–307, 1979.
- [30] Catherine Tuglus and Mark Laan. Modified fdr controlling procedure for multi-stage analyses. *Statistical applications in genetics and molecular biology*, 8:Article 12, 01 2009.
- [31] Zaiwen Wen, Donald Goldfarb, and Wotao Yin. Alternating direction augmented lagrangian methods for semidefinite programming. *Mathematical Programming Computation*, 2(3):203–230, Dec 2010.
- [32] Zhang Z. *The Schur Complement and Its Applications*. 01 2005.