UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Master's thesis
(Magistrsko delo)

Ridge regression for categorical data

(L2 regresija za opisne podatke)

Ime in priimek: Marko Palangetić
Študijski program: Matematične znanosti, 2. stopnja
Mentor: doc. dr. Rok Blagus

Koper, september 2018

Ključna dokumentacijska informacija

Ime in PRIIMEK: Marko PALANGETIĆ

Naslov magistrskega dela: L2 regresija za opisne podatke

Kraj: Koper

Leto: 2018

Število listov: 66 Število slik: 17 Število tabel: 3

Število referenc: 18

Mentor: doc. dr. Rok Blagus

Ključne besede: Logistična regresija, James - Stein cenilka, L2 penalizacija

Math. Subj. Class. (2010): 62J07

UDK: 519.22(043.2)

Izvleček:

V magistrskem delu se ukvarjamo s problemom izboljšanja lastnosti standardne cenilke lokacijskega parametra v Bernoullijevi porazdelitvi. Kriterij za izboljšanje cenilke je srednja kvadratna napaka (SKN). Stein je predlagal izboljšano cenilko za ocenjevanje lokacijskega parametra za multivariatno normalno porazdelitev, v magistrskem delu pa si želimo rezultat posplošiti na Bernoullijevo porazdelitev. Glavna metoda, ki smo jo uporabili v ta namen, je penalizirana logistična regresija z uporabo L2 penalizacijske funkcije. Motivacija za praktični del magistrskega dela je, da nadgradimo rezultat, ki ga je Brown podal v članku o nastopih igralcev pri igri baseball. Prvi uporabljen pristop je, da predstavimo SKN kot funkcijo penalizacijskega koeficienta in z uporabo različnih optimizacijskih metod, poiščemo koeficient, ki nam bo podal boljši SKN. Drugi pristop je, da uporabimo metode prečnega preverjanja. To naredimo tako da poiščemo penalizacijski koeficient, ki optimizira kriterijsko funkcijo pridobljenjo s pomočjo prečnega preverjanja, kjer uporabljamo različne kriterijske funkcije. Na koncu sta oba pristopa empirično testirana.

Key words documentation

Name and SURNAME: Marko PALANGETIĆ

Title of the thesis: Ridge regression for categorical data

Place: Koper

Year: 2018

Number of pages: 66 Number of figures: 17 Number of tables: 3

Number of references: 18

Mentor: Assist. Prof. Rok Blagus, PhD

Keywords: Logistic Regression, James - Stein estimator, Ridge penalization

Math. Subj. Class. (2010): 62J07

UDK: 519.22(043.2)

Abstract:
In the master thesis, we are considering the problem of improving the standard estimator of the location parameter in Bernoulli distribution. Criteria for improving the estimator is the Mean Squared Error risk. Stein have provided improved estimator for the location parameter of multivariate normal distribution. In the thesis, we want to extend the result on Bernoulli distribution. The main method used for that purpose is the logistic regression with ridge penalization. A motivation for the practical part of the thesis is to improve the result from the Brown's article about batting averages. First approach was to express the MSE as a function of the penalization coefficient, and using various optimization techniques, to find a coefficient which gives a better MSE. The second approach was to use leave-one-out cross validation methods. That is used such that we calculate the penalization coefficient which optimize cross validation objective function. Here we used different objective functions. At the end, both approaches are empirically tested.

Acknowledgement

Velika zahvala gre mojemu mentorju, doc. dr. Roku Blagusu za njegove smernice, koristno pomoč in strokovne nasvete pri izdelavi magistrskega dela. Posebna zahvala gre Fakulteti za matematiko, naravoslovje in informacijske tehnologije za izkazano podporo s štipendijo in finančno pomoč za udeležbo na različnih mednarnodnih tekmovanjih in konferencah.

Превасходно бих се желио захвалити мојим родитељима, оцу Миши и мајци Вукосави, за несебичну подршку током цјелокупног школовања. Посебно се захваљујем мојој дјевојци Слађани за помоћ при доношењу исправних одлука у критичним тренуцима.

Contents

List of Tables

List of Figures

List of Abbreviations

i.e.	that is
e.g.	for example
i.i.d.	Independent and identicaly distributed
MSE	Mean squared error
GMSE	Generalized mean squared error
LOOCV	Leave-one-out cross validation
PDF	Probability distribution function
ML	Maximum likelihood

1 Introduction

1.1 Overview on data analysis

Data is becoming a fundamental resource in the modern world. Production of data increased rapidly in the recent period, and due to that, we would like to analyze it in order to try to improve some aspects of our reality. That analysis can be done by many mathematical disciplines. Different mathematical disciplines use different approaches for modeling problems. Those disciplines were changing their names during time, but today, the most important disciplines, when it comes to data, are Statistics, Machine Learning and Artificial Intelligence. However, in all those fields, the main assumption is that data is a realization of a random variable. Therefore, modeling problem is moved to the field of the probability theory, which is the core of all above mentioned disciplines. Because of that assumption, we would like to know more about the unknown random variable which produced our data. The main characteristic of a random variable is the distribution with respect to Lebesgue measure on some Euclidean space, if the variable is continuous. If the variable is discrete, distribution with respect to the counting measure is taken. So, one of the main tasks is to identify that distribution. There are many methods for that, and they can be divided into two classes: parametric and non-parametric. Parametric methods are those for which we assume that the unknown distribution is coming from family of distributions, which is determined by a finite number of numerical parameters. Therefore, to identify the unknown distribution means to identify the unknown parameters. Examples of such methods are maximum likelihood method and method of moments. On the other side, non-parametric methods do not have such assumption. An example of a non-parametric method is kernel density estimation, where we are using building block functions called kernels to construct the probability distribution function. In this thesis, we will focus on the parametric methods where we will look into particular properties of a parameter estimation.

1.2 Parameter estimation problem

Determination of unknown parameters in a family of parametric distributions is called estimation. Let us define it more formally. Let $(\Omega, \mathcal{F}, \mathbb{P}_\theta)$ be a probability space, with

an unknown probability measure, dependent on the unknown parameter $\theta \in \Theta$. Let $\mathcal{X}$ be a measurable space, and let $X : \Omega \to \mathcal{X}$ be a random variable. Our goal is to determine the unknown parameter θ, based on the random variable X. We define an estimator as a function $\hat{\theta} : \mathcal{X} \to \Theta$, which is in fact our unknown value. For a particular realization of the variable X, we want to have an estimation of the parameter θ. In the case when we have given i.i.d. sample of the size n, the estimator is of the form $\hat{\theta} : \mathcal{X}^n \to \Theta$. The most used method for the parameter estimation, mentioned above, is the maximum likelihood method. However, after the estimation process, we would like to know how good our estimator is. One of the useful properties of a good estimator is unbiasedness. An estimator is unbiased when $E_\theta(\hat{\theta}(X)) = \theta$, which is, the expectation of the estimator is equal to the actual value of the parameter. For the other quality indicators we will define a loss function, which plays a role of a cost for wrong estimation. The loss function is of the form $l : \Theta \times \Theta \to \mathbb{R}^+$. So, to benefit from the estimator, we would like to have the least possible risk. Risk is defined as:

$$R(\hat{\theta}, \theta) = E_\theta(l(\theta, \hat{\theta}(X))).$$

The risk is defined with respect to a loss function. We provide two definitions.

Definition 1.1. An estimator $\hat{\theta}$ *dominates* an estimator $\hat{\theta}^*$ with respect to the risk $R(\cdot, \theta)$ if $R(\hat{\theta}^*, \theta) \leq R(\hat{\theta}, \theta)$ for all θ, and the inequality is strict for some θ.

Definition 1.2. An estimator $\hat{\theta}$ is called *admissible* with respect to the risk function R, when there is no estimator that dominates it with respect to R; otherwise it is inadmissible.

Here, the term "admissibility" is used as a sort of minimality.
For our practical problems, we will have that $\Theta = \mathbf{R}^m$ and the loss function will be

$$l(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}}) = \frac{1}{m}(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})^T(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}). \tag{1.1}$$

From now on, we will use bold notation for vectors. The risk with respect to the loss function (1.1) is called mean squared error (MSE). We see its importance from the decomposition which can be done in one-dimensional space. Let $\hat{\theta}$ be an estimator of the parameter θ. We write:

$$\begin{aligned} E((\hat{\theta} - \theta)^2) &= E((\hat{\theta} - E(\hat{\theta}) + E(\hat{\theta}) - \theta)^2) \\ &= E((\hat{\theta} - E(\hat{\theta}))^2 - 2(\hat{\theta} - E(\hat{\theta}))(E(\hat{\theta}) - \theta) + (E(\hat{\theta}) - \theta)^2) \\ &= E((\hat{\theta} - E(\hat{\theta}))^2) - 2E(\hat{\theta} - E(\hat{\theta}))(E(\hat{\theta}) - \theta) + (E(\hat{\theta}) - \theta)^2 \\ &= E((\hat{\theta} - E(\hat{\theta}))^2) + (E(\hat{\theta}) - \theta)^2. \end{aligned} \tag{1.2}$$

In the expression (1.2), the first term is the variance of the estimator $\hat{\theta}$, while the second one without a square is called bias. Bias represents the distance of the expectation from the real parameter value. The interpretation of those two terms is very important; their trade-off is crucial in data analysis. Huge bias implies that the estimator missed relevant properties of the data, and it did not recognized the real distribution of the data. This phenomena is called underfitting. For example, it can appear when the actual distribution of the data is extremely different than any distribution from the parametric family. On the other side, huge variance is telling us that the small changes in the data may cause completely different estimation (instability). This phenomena is called overfitting. In the further chapters we will talk more about underfitting and overfitting. Provided interpretation of the values coming from the MSE decomposition, is showing us the relevance of the MSE as a risk. For an unbiased estimator, the bias term is equal to 0. Therefore, the MSE of an unbiased estimator is equal to the variance.

1.3 Stein paradox

As we can see from the definition (1.2), an admissible estimator represents an estimator with the best performance with respect to some risk. So, we are asking ourselves, for the provided risk, in our case MSE, is there an admissible estimator. Usually, we add a constraint that we want to have an admissible unbiased estimator. In this case, we are looking for an unbiased estimator with the smallest possible variance. Well-known results from that topic may be found in the articles [12] and [7]. A well known framework for a construction of such estimators is to use sufficient statistics. The question is, what is happening when we do not take unbiased property as our assumption. One of the first results from this topic was given by Stein in the article [16]. For his problem, he assumed that we have data which is coming from the multivariate normal distribution of the dimension m. Here, we have unknown mean $\boldsymbol{\mu}$ and known covariance matrix of the form $\sigma^2 I$. σ is known constant, while I stands for the identity matrix. From the construction, we can see that he assumed independence between the components of the normal vector. For a given sample point $\mathbf{X}_1$ from that distribution, the ML method will give us an estimator $\hat{\boldsymbol{\theta}} = \mathbf{X}_1$. Using Cramer-Rao inequality from [2], it can be shown that the obtained estimator is unbiased with the least possible variance. On the other side, for $m = 1$ or $m = 2$, it is shown, for example in [7], that the ML estimator is admissible with respect to MSE (taking also biased estimators into the consideration). Stein has shown that for $m \geq 3$, the admissibility does not hold. As a counter example, he constructed an estimator, the so-called James-Stein

estimator:

$$\hat{\boldsymbol{\theta}}_{JS} = \left(1 - \frac{(m-2)\sigma^2}{\|\mathbf{X}_1\|^2}\right)\mathbf{X}_1,$$

where $\|\cdot\|$ stands for the 2-norm of a vector. Since the unbiased estimators were considered as those with the best properties, constructing an estimator with smaller MSE than the MSE of an unbiased estimator was unexpected. Therefore, the existence of such estimator is called Stein's paradox. If we have a sample of size n, and we denote $\mathbf{X} = \frac{\bar{\sum}_{i=1}^{n} \mathbf{X}_i}{n}$, then James-Stein estimator can be formulated as:

$$\hat{\boldsymbol{\theta}}_{JS} = \left(1 - \frac{(m-2)\sigma^2}{\|\bar{\mathbf{X}}\|^2}\right)\bar{\mathbf{X}},$$

There is a more intuitive interpretation of the phenomena. Given that all components of the multivariate normal distribution are independent, we can analyze them separately. So, taking each component for itself, we have that the ML estimator for each component is admissible. On the other side, taking them all together, we have a better estimator than ML one. This shows that combined analysis of the independent events, may give us better result than the individual analysis.

1.4 Problem formulation

Our goal is to extend Stein's result on the Bernoulli distribution. We say that a random variable Y has the Bernoulli distribution with a parameter π if it has two possible outputs, 0 and 1, where probability for output 1 is π. The distribution may be represented as:

$$P(Y = y) = \pi^y(1-\pi)^{1-y}, \quad y \in \{0, 1\}.$$

The notation for this is $Y \sim \text{Bern}(\pi)$. The other properties are: $E(Y) = \pi$ and $\text{Var}(Y) = \pi(1-\pi)$. So, we will assume that we have data coming from K different Bernoulli distributions, and we will try to show that estimating them together, will give a better MSE than the average MSE of the particular estimations. We provide the analysis for the individual case. Let $Y_1, \ldots, Y_n$ be an i.i.d. sample with a Bernoulli distribution with unknown parameter π, and let $y_1, \ldots, y_n$ be the realization of the sample. By the MLE principle, we want to maximize the probability $P(Y_1 = y_1, \ldots, Y_n = y_n)$ which we call likelihood. Due to independence, we know that the likelihood is:

$$L(\pi) = P(Y_1 = y_1, \ldots, Y_n = y_n) = \prod_{i=1}^{n} \pi^{y_i}(1-\pi)^{1-y_i} = \pi^a(1-\pi)^b,$$

where a is the number of ones in the sample, while b is the number of zeros. For many cases when we use the ML method, it is easier to maximize the logarithm of the

likelihood, denoted with $l(\pi)$. We have that

$$l(\pi) = a \log \pi + b \log(1 - \pi).$$

Shorter, we call it log-likelihood. Maximizing over π, we get the explicit expression for the estimator:

$$\hat{\pi} = \frac{a}{a+b} = \frac{\sum_{i=1}^{n} Y_i}{n},$$

which is again the average, the same estimator as for the location parameter of the normal distribution. To check if it is unbiased, we have the following:

$$E(\hat{\pi}) = \frac{\sum_{i=1}^{n} E(Y_i)}{n} = \frac{\sum_{i=1}^{n} E(Y_1)}{n} = E(Y_1) = \pi.$$

The estimator is unbiased, so the MSE is equal to the variance. To calculate MSE we have that:

$$\mathrm{Var}(\hat{\pi}) = \frac{\sum_{i=1}^{n} \mathrm{Var}(Y_i)}{n^2} = \frac{\sum_{i=1}^{n} \mathrm{Var}(Y_1)}{n^2} = \frac{\mathrm{Var}(Y_1)}{n} = \frac{\pi(1-\pi)}{n}.$$

If we have K different, independent, Bernoulli distributed random variables, we can organize their parameters into a vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^T$, while their particular ML estimation we organize into a vector $\hat{\boldsymbol{\pi}} = (\hat{\pi}_1, \ldots, \hat{\pi}_K)$. That vector is formed from samples of different sizes $a_{k\bullet}$, $k \in \{1, \ldots, K\}$. So the joint MSE will be

$$R(\hat{\pi}, \pi) = E(\frac{1}{K}(\hat{\boldsymbol{\pi}} - \boldsymbol{\pi})^T(\hat{\boldsymbol{\pi}} - \boldsymbol{\pi})) = \frac{1}{K}\sum_{k=1}^{K} E((\hat{\pi}_k - \pi_k)^2) = \frac{1}{K}\sum_{k=1}^{K} \frac{\pi_k(1-\pi_k)}{a_{k\bullet}}. \quad (1.3)$$

Our goal is to improve (1.3) by using methods of the logistic regression with ridge penalization.

1.5 Practical motivation

One of the aims of the thesis is to improve the result presented in the article [4]. Namely, the article is trying to solve the problem of the prediction of a performance of certain baseball players. For each individual player, we have collected the data from the first half-season, and we would like to predict the performance in the second half-season. The parameter we want to predict is the percentage of the successful hits for each player. Data which we have from the first half-season represents the total number of hits and the number of successful hits. We have those data for each player separately. From the second half-season, we have the total number of hits for each player. Our goal is to predict the number of the successful hits in the second half-season. Let us write the problem using the notations that we had before. Each hit is a Bernoulli distributed variable with values 1-successful and 0-unsuccessful, with an

unknown probability which we are supposed to estimate. That probability is different from player to player. So, the hits may be interpreted as the data coming from different, independent Bernoulli distributions. Here we take a reasonable assumption that the performances of the players are independent. Here, probability is a parameter which measures the performance of the particular player. Our goal is to estimate it in order to estimate the performance. We provide the brief description of the approach described in the original article. Let N_{ji} be the total number of hits in the half-season j, $j = 1, 2$ of the player i, $i \in \{1, \ldots, P\}$. Here, P is the total number of players. Let H_{ji} be the number of the successful hits with the same index description. We have that

$$H_{ji} \sim Bin(N_{ji}, p_i).$$

For the ratio of the successful hits we have $R_{ji} = \frac{H_{ji}}{N_{ji}}$. This ratio, for $j = 1$, is exactly the unbiased estimator of the probability p_i for each player. The goal was not just to end here, but to try to improve the quality of the estimation, based on certain risk functions. For that purpose, the author used the fact that R_{ji} has nearly normal distribution with the mean p_i [3]. This assumption is justified for large N_{ji} and when p_i is close to 0.5. The variance, in such normal distribution, is dependent on the unknown p_i. This was not satisfiable, so the author introduced the variance-stabilizing transformation. He created a new random variable as:

$$X_{ji} = \arcsin \sqrt{\frac{H_{ji} + 1/2}{N_{ji} + 1/4}}.$$

For such random variable, we have that it has nearly normal distribution with $E(X_{ji}) = \arcsin \sqrt{p_i}$ and $\mathrm{Var}(X_{ji}) = \frac{1}{4N_{ji}} + O(\frac{1}{N_{ji}^2})$. Big O is a notation that describes the order of the remaining part. So, using the described transformation, the author obtained nearly normal random variable for which the variance is independent from the unknown parameter p_i. So he satisfied the assumptions to use the James - Stein estimator. Beside the James - Stein estimator, the author used many other methods to estimate the unknown mean of X_{1i}. From those results, he constructed the estimator for the probability p_i using the inverse transformation. To evaluate the performance of the estimators, he used the loss function called total squared error (TSE). It is derived as:

$$TSE(\tilde{R}) = \sum_{i=1}^{P} (R_{2i} - \hat{R}_i)^2 - \sum_{i=1}^{P} \frac{R_{2i}(1 - R_{2i})}{N_{2i}}, \tag{1.4}$$

where $\hat{R}_i$ is the estimator of the probability p_i. This TSE risk is based on the out-of-sample MSE, about which we will talk later. It represents MSE of the data from the testing set, decreased by the estimation of the variance of the test data.

So, our goal is to improve the results provided in the article, hoping that we will achieve a better TSE than the methods from the article.

1.6 Structure of the thesis

This master thesis consists of six chapters including this one. Chapter 2 provides a theoretical setup of the prediction problem, basic results from statistical learning theory and an explanation of those results on the example of logistic regression. At the end of the chapter, we provide a logistic regression model for our problem described above and probability estimators obtained using the model. In Chapter 3, we introduce penalization as a tool for overfitting prevention. We introduce LASSO and ridge penalizations and we describe their benefits for linear models. Further, we use ridge penalization for our problem of probability estimation, where we construct different estimators for different versions of the ridge penalization. We provide basic properties of such estimators. In the next chapter, we talk about cross validation. We provide a basic explanation about the topic and also how it can be used for our problem. We use cross validation to calculate the penalizing coefficient in our estimators. In Chapter 5 we have a simulation study. Since we were unable to prove many properties theoretically, we test them using simulations. We test estimators for some special cases and also for the general case. At the end of the chapter, we provide the result obtained using the data from the Brown's article. We compare it with the result from the article. In the final chapter we summarize the obtained results and we give some ideas and plans for a future research.

2 Prediction problem and Logistic regression

In various aspects of science we have many quantities that are measured. In many cases, we are interested to know how one quantity affects the other, or how a set of quantities affects another set. For example in economy, we may be interested in the impact of the GDP growth on the unemployment rate, or how the amount of the cars in a particular city is related to the CO_2 concentration in the air. If we want to completely explain one quantity Y using a set of quantities X, we say that we want to predict Y using X. Such problem is called prediction problem or supervised learning problem. The main assumption in modeling of the measured quantities is that we assume that they are a realization of an unknown random variable. Using that assumption, we can say that we want to explain one random variable using other one. Putting that into more formal framework, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. We define two random variables $X : \Omega \to \mathcal{X}$ and $Y : \Omega \to \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are some measurable spaces, usually euclidean ones. We assume that the pair (X, Y) is distributed with unknown probability distribution $\mathcal{P}$, $(X, Y) \sim \mathcal{P}$. We define a function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ which we call loss function. This function represents a cost which we are ready to pay for a wrong prediction. If we have a perfect prediction, we will not have to pay any cost, so we add a constraint that $l(y, y) = 0$ for every $y \in \mathcal{Y}$. Our goal is to find a measurable function $f^* : \mathcal{X} \to \mathcal{Y}$ such that:

$$f^*_{\mathcal{P}} = \arg\min_f E_{\mathcal{P}}(l(Y, f(X))).$$

Variable X is called the predictor, while Y is called the target. The function f is named oracle or Bayes predictor. If $\mathcal{Y} = \mathbb{R}$, then we are talking about a regression problem, while if $|\mathcal{Y}| < \infty$ then we face a classification problem. In the special case when $|\mathcal{Y}| = 2$, we have a binary classification problem. That problem is quite well studied in the literature and it will be the problem of interest for us. Here, brackets $|\cdot|$ applied on a set stand for the cardinality of that set. For a certain function f, the quantity $R_{\mathcal{P}}(f) = E_{\mathcal{P}}(l(Y, f(X)))$ is called risk. Risk of the Bayes estimator is called Bayes risk, and we denote it as $R^*_{\mathcal{P}} = R_{\mathcal{P}}(f^*_{\mathcal{P}})$. The difference between the risk and the Bayes risk, $R_{\mathcal{P}}(f) - R^*_{\mathcal{P}}$, is called excess risk.

2.1 Regression

Now, consider the case when we have a regression problem. First we need to decide which loss function to choose in that case. One very intuitive case is to take a distance between values: $l(y, \bar{y}) = |y - \bar{y}|$. This function is called Mean Absolute Error (MAE). In practice, it can be computationally very hard to work with absolute values, because functions with absolute values are not differentiable at some points. So, the most widely used loss function for the regression problem is Mean Squared Error (MSE): $l(y, \bar{y}) = (y - \bar{y})^2$. Using MSE, our problem becomes the minimization of $E_{\mathcal{P}}((Y - f(X))^2)$ over the set of the measurable functions f.

Lemma 2.1. *For the risk defined as $R_{\mathcal{P}}(f) = E_{\mathcal{P}}((Y - f(X))^2)$, the Bayes predictor is $f^*_{\mathcal{P}}(x) = E(Y|X = x)$.*

Proof. We have that:

$$\begin{aligned} E((Y - f(X))^2) &= E(E((Y - f(X))^2|X)) = E(E(Y^2 - 2Yf(X) + f(X)^2|X)) \\ &= E(E(Y^2|X) - 2E(Y|X)f(X) + f(X)^2) \\ &= E(E(Y|X)^2 - 2E(Y|X)f(X) + f(X)^2 + E(Y^2|X) - E(Y|X)^2) \\ &= E((E(Y|X) - f(X))^2 + E((Y - E(Y|X))^2|X)) \\ &= (E(Y|X) - f(X))^2 + E((Y - E(Y|X))^2). \end{aligned}$$

From the last expression we see that the minimum is achieved when $f(X) = E(Y|X) \Leftrightarrow f(x) = E(Y|X = x)$. □

2.2 Binary Classification

. The problem of binary classification will be of interest through the whole thesis. Since $|\mathcal{Y}| = 2$, without loss of generality, we can encode $\mathcal{Y} = \{-1, 1\}$. A popular encoding is also when we replace -1 with 0, but to have shorter expressions in this section we will use encoding with -1. The most widely used loss function in this case is accuracy measure. It measures the percentage of wrongly classified instances. If we are estimating also probability of a correct classification, then there are other loss functions, like binary crossentropy or area under the ROC curve. In this case, we restricts ourselves only to the accuracy loss. We defile our loss function as: $l(y, \bar{y}) \neq \mathbf{1}\{y = \bar{y}\}$. Optimization problem becomes minimization of $E(\mathbf{1}\{Y \neq f(X)\}) = P(Y \neq f(X))$.

Lemma 2.2. *For the risk defined as $R_{\mathcal{P}}(f) = E(\mathbf{1}\{Y \neq f(X)\})$, the Bayes predictor is $f^*(x) = \mathbf{1}\{P(Y = 1|X = x) \geq \frac{1}{2}\}$.*

Proof. Let us define $\bar{f}(x) = \mathbf{1}_{\{P(Y=1|X=x)\geq\frac{1}{2}\}}(x)$. Let f be an arbitrary prediction function $f : \mathcal{X} \to \mathcal{Y}$. We want to show that $R(f) \geq R(\bar{f})$. We have that:

$$\begin{aligned} R(f) &= E(\mathbf{1}\{Y \neq f(X)\}) = E(E(\mathbf{1}\{Y \neq f(X)\}|X)) \\ &= \int E(\mathbf{1}\{Y \neq f(X)\}|X = x)P_x(dx) \\ &\geq \int E(\mathbf{1}\{Y \neq \bar{f}(X)\}|X = x)P_x(dx) = R(\bar{f}). \end{aligned}$$

The remaining question is how to prove the part $E(\mathbf{1}\{Y = f(X)\}|X = x) \geq E(\mathbf{1}\{Y = \bar{f}(X)\}|X = x)$. We have that:

$$\begin{aligned} E(\mathbf{1}\{Y \neq f(X)\}|X = x) &\geq \min_{a\in\{0,1\}} E(\mathbf{1}\{Y \neq a\}|X = x) = \min_{a\in\{0,1\}} P(Y \neq a|X = x) \\ &= \begin{cases} P(Y \neq 1|X = x) & \text{if} \quad P(Y \neq 1|X = x) < P(Y \neq 0|X = x) \\ P(Y \neq 0|X = x) & \text{otherwise} \end{cases} \\ &= \begin{cases} P(Y \neq 1|X = x) & \text{if} \quad P(Y \neq 1|X = x) < \frac{1}{2} \\ P(Y \neq 0|X = x) & \text{otherwise} \end{cases} \\ &= P(Y \neq \bar{f}(X)|X = x) = E(\mathbf{1}\{Y \neq \bar{f}(X)\}|X = x). \end{aligned}$$

With this we ended the proof. □

2.3 Relation between classification and regression

Sometimes in practice, it is quite easier to formulate a classification problem as a regression one from the various reasons. For example, accuracy loss is a non-convex function, which will cause non-convex optimization problems. Those problems are really hard to solve. So, the question which we ask ourselves is the following: if we have enough good solution for the regression problem, is it also a good solution for the classification problem. We formulate the following theorem.

Theorem 2.3. *Let X and Y be random variables like described before. Let $R_{cl}(f) = E(\mathbf{1}\{Y \neq f(X)\})$ and $R_{reg}(f) = E((Y - f(X))^2)$ represent classification and regression risk respectively. Let R^*_{cl} and R^*_{reg} represent respective Bayes risks, and let $\bar{\eta}(x)$ be an arbitrary regressor. Define $\bar{g}(x) = \mathbf{1}\{\bar{\eta}(x) \geq \frac{1}{2}\}$. Then it holds:*

$$R_{cl}(\bar{g}) - R^*_{cl} \leq 2\sqrt{R_{reg}(\bar{\eta}) - R^*_{reg}}.$$

Proof. Let $g*$ and $\eta*$ be Bayes classifier and Bayes regressor respectively. From the previous chapter we know that $g^*(x) = \mathbf{1}\{\eta^*(x) \geq \frac{1}{2}\}$. Notice that $R_{\mathrm{cl}}(\bar{g}) = E(\mathbf{1}\{Y \neq$

$\bar{g}(X)\}) = E((Y - \bar{g}(X))^2)$, because the difference can be only 0 or 1, so the expressions are equivalent. We have that

$$\begin{aligned} R_{\text{cl}}(\bar{g}) &= E((Y - \bar{g}(X))^2) = E(Y^2) - 2E(Y\bar{g}(X)) + \overbrace{E(\bar{g}(X)^2)}^{E(\bar{g}(X))} \\ &= E(Y^2) + E(\bar{g}(X)) - 2E(\bar{g}(X)E(Y|X))) \\ &= E(Y^2) + E(\bar{g}(X)) - 2E(\bar{g}(X)\eta^*(X)) \\ &= E(Y^2) + 2E(\bar{g}(X)(\frac{1}{2} - \eta^*(X))). \end{aligned}$$

Using analogy, for excess risk we have that:

$$R_{\text{cl}}(\bar{g}) - R^*_{\text{cl}} = 2E((\bar{g}(X) - g^*(X))(\frac{1}{2} - \eta^*(X))).$$

If $\bar{g}(x) > g^*_{cl}(x)$, then we must have $\bar{g}(x) = 1$ and $g^*_{cl}(x)$. Further that implies:

$$\bar{\eta}(x) > \frac{1}{2} \wedge \eta^*(x) < \frac{1}{2} \Rightarrow \bar{\eta}(x) > \frac{1}{2} > \eta^*(x).$$

On the other side, by analogy, condition $\bar{g}(x) < g^*_{cl}(x)$ implies that $\bar{\eta}(x) < \frac{1}{2} < \eta^*(x)$.
We define events: $A = \{x \,|\, \bar{g}(x) \geq g^*_{cl}(x)\}$, $B = \{x \,|\, \bar{g}(x) < g^*_{cl}(x)\}$ and $C = \{x \,|\, \bar{g}(x) = g^*_{cl}(x)\}$. We continue our calculations

$$\begin{aligned} R_{\text{cl}}(\bar{g}) - R^*_{\text{cl}} &= 2E(((\bar{g}(X) - g^*(X))(\frac{1}{2} - \eta^*(X))\mathbf{1}_A) \\ &\quad + 2E(((\bar{g}(X) - g^*(X))(\frac{1}{2} - \eta^*(X)))\mathbf{1}_B) \\ &\quad + 2E(((\bar{g}(X) - g^*(X))(\frac{1}{2} - \eta^*(X)))\mathbf{1}_C) \\ &= 2E((\frac{1}{2} - \eta^*(X))\mathbf{1}_A) + 2E((\eta^*(X) - \frac{1}{2}))\mathbf{1}_B) \\ &\leq 2E((\bar{\eta}(X) - \eta^*(X))\mathbf{1}_A) + 2E((\eta^*(X) - \bar{\eta}(X))\mathbf{1}_B) \\ &= 2E(|\bar{\eta}(X) - \eta^*(X)|) \leq 2\sqrt{E((\bar{\eta}(X) - \eta^*(X))^2)} \\ &= 2\sqrt{R_{\text{reg}}(\bar{\eta}) - R^*_{\text{reg}}}. \end{aligned}$$

The last inequality in the previous expression the Cauchy-Swartz one.

□

2.4 Empirical case

Given that X and Y are random variables, and that their probability distribution is unknown, the above described optimization problem is not solvable in reality. On

the other hand, in practice we have given the data pairs $(x_1, y_1), \ldots, (x_n, y_n)$ for some $n \in \mathbb{N}$. Those data represent realizations of the random pair (X, Y). The usual term we use for those data is sample, and the amount of those data we call the sample size. The idea here is to replace the probability distribution $\mathcal{P}$ with its empirical approximation. The empirical distribution of $z_i = (x_i, y_i)$ is

$$\hat{\mathcal{P}}_n(A) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{z_i \in A\}}.$$

Using the weak law of large numbers we have that

$$\frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{z_i \in A\}} \xrightarrow[n \to \infty]{} E_{\mathcal{P}}(\mathbf{1}_{\{(X,Y) \in A\}}) = \mathcal{P}(A),$$

where the above convergence is convergence in probability. It shows us that the approximation is consistent. Using analogy, we define the empirical risk using empirical distribution. We have that:

$$R_{\hat{\mathcal{P}}_n}(f) = E_{\hat{\mathcal{P}}_n}(l(Y, f(X))) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, f(x_i)).$$

Since we have something deterministic, the naive idea is to try to directly minimize $R_{\hat{\mathcal{P}}_n}(f))$, but any function which satisfies that $y_i = f(x_i)$ for every i, is an optimal solution. In that case, the empirical loss is zero. Denote such minimizer with f_n^*. The question is if any function from the set of minimizers can mimic f^*. The minimizer of the empirical risk is too dependent on the sample, and that dependency phenomena is called overfitting. To avoid overfitting, we will restrict our search space to some subspace of the functions $\mathcal{S}$. The problem becomes to find

$$f_{n,\mathcal{S}}^* = \arg\min_{f \in \mathcal{S}} \frac{1}{n} \sum_{i=1}^{n} l(y_i, f(x_i)). \tag{2.1}$$

Any minimizer from (2.1) is called the empirical risk minimizer. We are interested in how well the estimated risk minimizer $f_{n,\mathcal{S}}^*$ approximates Bayes predictor f^*. For that purpose, we decompose the excess risk $R(f^*) - R(f_{n,\mathcal{S}}^*)$:

$$R(f^*) - R(f_{n,\mathcal{S}}^*) = R(f^*) - R(f_n^*) + R(f_n^*) - R(f_{n,\mathcal{S}}^*).$$

The part $R(f^*) - R(f_n^*)$ is called a stochastic error (learning error, prediction error). The part $R(f_n^*) - R(f_{n,\mathcal{S}}^*)$ is named an approximation error. The art of a good prediction lies in balancing between those two errors. We distinguish two cases here:

- A stochastic error dominates an approximation error: this phenomena is overfitting mentioned above. In this case, our prediction is too much data dependent. It can lead to a wrong prediction if we apply our function $f_{n,\mathcal{S}}^*$, on a new sample point.

- An approximation error dominates a stochastic error: phenomena known as underfitting. In this case, we have that our set $\mathcal{S}$ is very restricted. If f^* is very far from $\mathcal{S}$, then $f^*_{n,\mathcal{S}}$, which is a member of $\mathcal{S}$, will be a very poor approximation of it.

In practice, it is really important to get rid of those two phenomenas. The best indicator of underfitting is a huge empirical risk, so it is easy for detection. Underfitting ocurrs usualy when we try to use very simple model (small set $\mathcal{S}$) on the huge and complex data. The overfitting detection is not so trivial, since in that case we can have very satisfiable empirical risk. The overfitting detection is done by dividing our sample into two sets, called train and test sets. The idea is to find an empirical risk minimizer by using the train set, and to apply it on the test set. Since both sets are samples from the same probability distribution, the calculated risks on the both sets should be almost the same. In practice those risks are usually denoted as a train error and a test error. Overfitting, which we characterized as a huge data dependency, appears when the train error is significantly larger than the test error. We interpret that in the following way: the empirical risk minimizer is so dependent on the train data that if we provide to the minimizer a new, unseen data, it will perform badly. Division of the sample on the train and test set should satisfy two conditions.

- The train set should still be very large, since a small train set may lead to overfitting.
- The test set should not be very small, since in that case, it is not a representative sample of the unknown probability distribution. A small test set may lead to an empirical distribution that is very different from the real one. That further leads to a large test error.

The usual division of a sample on the train and the test set is done such that the ratio of their sizes is $5 : 1$ or $10 : 1$.

2.5 Modeling

There are two main modeling approaches for prediction problems: generative modeling and discriminative modeling. We provide the main differences.

- **Generative modeling**

  In the previous part of the thesis, we have constructed minimizers for the classification and regression risks, called bayes predictors. So in practice, one of the possibilities is to try to estimate bayes predictor which is the optimal solution.

That approach is called generative modeling. For the regression problem, we would like to estimate $E(Y|X = x)$ as a function of x. In the classification case, the bayes predictor is the function of $E(Y|X = x)$. So, if $\bar{\eta}(x)$ is a good estimator of $E(Y|X = x)$, then also $\bar{g}(x) = \mathbf{1}\{\bar{\eta}(x) \geq \frac{1}{2}\}$ is a good approximation for the Bayes classifier according to the theorem 2.3. In both cases, our goal is to find a proper estimation of $E(Y|X = x)$. Usually, here we assume that $E(Y|X)$, as a random variable, is coming from the parametric family of distributions. Therefore, the estimation is usually done using the maximum likelihood method.

- **Discriminative modeling**

  In the previous part, we have also introduced empirical risk, based on the sample. So instead of estimating the minimum, as it was in the generative approach, we are minimizing the empirical risk. The empirical risk may be seen as an estimatior of the risk. For the regression problems, we usually define set $\mathcal{S}$ as some parametric family of functions. Then, we are trying to optimize the empirical loss, which is in the regression case the average of the quadratic losses. Such minimization is differentiable and convex optimization problem. In the case of the classification, accuracy loss is non-differentiable and non-convex function. So the optimization is usually done by replacing the accuracy loss with suitable convex, differentiable upper bound function. After replacing, we optimize the empirical risk using the upper bound loss.

Process of replacing the accuracy function with a convex upper bound is called convexification. We will show that in some cases, there exists a theoretical guarantee for that. For this purpose, we will need to encode our dependent variable Y with labels $\{1, -1\}$, instead of $\{1, 0\}$. Then, the accuracy loss may be written as $l(y, \bar{y}) = \mathbf{1}_{\{y\bar{y}<0\}}$. We rewrite our loss function as $l(y, \bar{y}) = g(y\bar{y})$ for $g(x) = \mathbf{1}_{\{x<0\}}$. The idea, which is used in many known algorithms, is to replace the function g, as a function of one variable, with some convex and differentiable h. After we repplaced g with h, we may write the loss as $l(y, \bar{y}) = h(y\bar{y})$. The next result from [1] give us a theoretical guarantee for such approach.

Theorem 2.4. *Let $h : \mathbb{R} \to \mathbb{R}_+$ be non-increasing, differentiable at 0 with $h'(0) < 0$. Let $R(f) = E(\mathbf{1}_{\{Yf(X)<0\}})$ be the classification risk, and $R_h(f) = E(h(Yf(X)))$ be the risk after the convexification called h-risk. Let f^* be a measurable function such that:*

$$R_h(f^*) = \min_f R_h(f) = R_h^*.$$

Then we have:

$$R(f^*) = \min_f R(f) = R^*.$$

Theorem (2.4) shows us that if we know the distribution $\mathcal{P}$, then optimizing h-risk is a good idea even if we are interested in the accuracy risk. More generally, the same authors showed that if $R_h(f) - R_h^*$ is small, then also $R(f) - R^*$ is. The choice of different functions h define different models. Famous models for generative approach are generalized linear models, kernel methods, k-nearest neighbors, Naive Bayes, Tree based methods, etc. Models that are using discriminative approach are Support Vector Machine, Support Vector Regression, Neural networks, Boosting approaches and so on. These two approaches are not completely disjoint, and many models may be interpreted using both approaches. One of those examples is logistic regression. In the next section we will see how the logistic regression can be derivate from both approaches.

2.6 Logistic regression

Logistic regression is one of the oldest models used for binary classification, with still very high usage in practice. Even though there are more powerful models which are performing with smaller errors, logistic regression is used because of its simplicity and interpretability. That will be the model which we will use for our problem, and a great example of the theory developed in the previous sections. The model is a part of a wider family of generalized linear models.
We introduce the sigmoid function:

$$S(x) = \frac{1}{1 + e^{-x}}.$$

We present the shape of the function in the Figure 1.

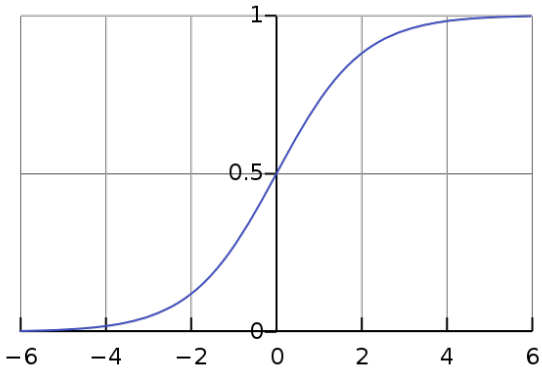


Figure 1: Sigmoid function

The domain of the function is the whole real line, while the range is the interval $[0, 1]$. This function is used mainly because of its characteristic shape, and the fact that it is not very difficult for optimization. From the shape, we can see that the function is in

one part very close to 0, and in the other very close to 1, which is a useful property for the problem of binary classification.

2.6.1 Generative approach

First we show the generative approach, modeling $E(Y|X)$. Let $\mathbf{X} = (X_1, \ldots, X_m)$ be a real random vector for some natural m, called feature vector. Let $Y \in \{0, 1\}$ be a Bernoulli random variable. We formulate a logistic regression model:

$$E(Y|\mathbf{X}) = P(Y = 1|\mathbf{X}) = S(\beta^T\mathbf{X} + b) = S(\beta_1 X_1 + \cdots + \beta_m X_m + b). \tag{2.2}$$

In the contest of the generative approach, the right side of the equation can be interpreted as a parametric family of distributions with parameters $(\beta, b) = (\beta_1, \ldots, \beta_m, b)$ which need to be determined. As we already mentioned, the method used for estimation is the maximum likelihood. Suppose that we have a sample of size n of pairs $(Y, \mathbf{X})$, denoted with $(Y_1, \mathbf{X}_1), \ldots, (Y_n, \mathbf{X}_n)$ and their realizations with $(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)$. For particular sample, we have that:

$$P(Y_i = y_i|\mathbf{X}_i = \mathbf{x}_i) = S(\beta^T\mathbf{x}_i + b)^{y_i}(1 - S(\beta^T\mathbf{x}_i + b))^{1-y_i}.$$

By the assumption, the sample points are independent among themselves, so their joint distribution is a product of marginals. We have that

$$P(Y_1 = y_1, \ldots, Y_n = y_n|\mathbf{X}_1 = \mathbf{x}_1, \ldots, \mathbf{X}_n = \mathbf{x}_n) = \prod_{i=1}^{n} S(\beta^T\mathbf{x}_i+b)^{y_i}(1-S(\beta^T\mathbf{x}_i+b))^{1-y_i}.$$

By the maximum likelihood method, our goal is to maximize the probability of the event that the sample is equal to its realization. In this case, it is conditional probability, and our goal is to maximize it. The last expression we denote with $L(\beta, b)$. For easier optimization, we introduce log-likelihood $l(\beta, b) = \log L(\beta, b)$. Hence, the previous expression becomes:

$$\begin{aligned} l(\beta, b) &= \sum_{i=1}^{n}(y_i \log S(\beta^T\mathbf{x}_i + b) + (1 - y_i)(1 - \log S(\beta^T\mathbf{x}_i + b))) \\ &= \sum_{i=1}^{n}(y_i \log(\frac{1}{1 + e^{-\beta^T\mathbf{x}_i+b}}) + (1 - y_i)(1 - \log(\frac{1}{1 + e^{-\beta^T\mathbf{x}_i+b}}))) \\ &= -\sum_{i=1}^{n}(y_i \log(1 + e^{-\beta^T\mathbf{x}_i+b}) + (1 - y_i)(1 - \log(1 + e^{-\beta^T\mathbf{x}_i+b})). \end{aligned}$$

Instead of maximizing previous quantity, the usual approach is to minimize its negation. So, the negation of the last expression is called negative log-likelihood, and we denote it as $nl(\beta, b)$. Since it is not possible to solve the equation $\frac{d\,nl(\beta)}{d\,(\beta,b)} = 0$ explicitly, we are

unable to obtain the exact expression for the vector (β, b). So, for this optimization problem numerical methods like gradient descent are used. About those methods, we will talk in the further sections.

Another interpretation of the negative log-likelihood, multiplied with the factor $\frac{1}{n}$, is in the form of an empirical risk. We define the loss function as:

$$l(y, \bar{y}) = -y \log(\bar{y}) - (1 - y) \log(1 - \bar{y}). \tag{2.3}$$

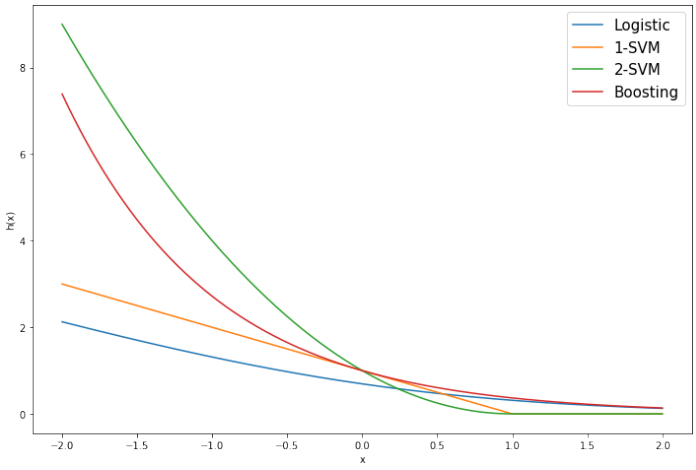
Then, the negative log-likelihood can be written as

$$\frac{1}{n} nl(\beta, b) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, S(\beta^T \mathbf{x}_i + b)).$$

The defined loss is called log loss, binary cross entropy or deviance. It is used with classification methods which are able to estimate the probability of particular class. Methods based on generalized linear models are using it widely.

2.6.2 Discriminative approach

As we mentioned above, the goal in the discriminative approach is to find an upper bound function for the accuracy loss such that it is convex and differentiable. Different candidates for upper bound determine different models. On the Figure 2, we provide a few candidate functions and the corresponding models which are using them.



Model	$h(x)$
Logistic regression	$\log(1 + e^{-x})$
Support vector machine (1 - SVM)	$\max(1 - x, 0)$
Support vector machine (2 - SVM)	$\max(1 - x, 0)^2$
Boosting	e^{-x}

Figure 2: Comparison of the different upper bound functions

The function of our interest is the logit function, used to construct the logistic regression. The logit function is defined as:

$$h(x) = \log(1 + e^{-x}).$$

Logit function satisfies the conditions of the Theorem 2.4, so we have the theoretical base for its use. The prediction function in this case is a scalar product mapping of the given vector $\mathbf{X} = (X_1, \ldots, X_n)$ as $\beta^T\mathbf{X} + b$. Using that, the Empirical risk for logistic regression is:

$$R(\beta) = \frac{1}{n}\sum_{i=1} h(\beta^T\mathbf{x}_i + b) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{-(\beta^T\mathbf{x}_i + b)}). \tag{2.4}$$

This loss is constructed with the variable Y encoded with $\{-1, 1\}$. With a little bit of algebra, it can be shown that (2.4) is equivalent to $\frac{1}{n}nl(\beta)$ described in the previous section, which was encoded with $Y \in \{-1, 1\}$. So optimizing different expressions for the different encodings give us the same result.

2.7 Discrete predictor

We will now set up a solution for our problem using logistic regression approach. As we already mentioned, we are assuming that our data is coming from K different Bernoulli variables; call them classes. We model them as a pair of random variables (Y, X), where X is taking values among K different discrete values, while Y, conditionally on X, has a Bernoulli distribution i.e. $P(Y = 1|X = k) = \pi_k$. To estimate unknown probabilities, we set up a logistic regression framework, assuming that X is predictor a and Y is target variable. Up to now, we have developed methods where independent variable X is coming from the euclidean space. In this case, variable X is taking finite number of values, for which a priori we do not have ordering. Modeling with $E(Y|X) = S(\beta X + b)$ is impossible in this case since X is not numerical. So we need to encode X as set of numerical variables such that we can use logistic regression for modeling. Different encodings give us different estimations. We define random variables $X_i = \mathbf{1}_{\{X=i\}}$, $i \in \{1, \ldots, k\}$ and a random vector $\mathbf{X} = (X_1, \ldots, X_k)$. Random vector $\mathbf{X}$ is a numerical equivalent of the categorical variable X, and we can use it for logistic regression modeling. Later on, we will show other possible encodings, which will give us different estimations. For this purpose we will use $\{0, 1\}$ encoding of the variable Y. As before, we assume that we are given a sample $X_1, \ldots, X_n$ and its realizations $x_1, \ldots, x_n$. We encode all of them as shown before, so we have random vectors $\mathbf{X}_1, \ldots, \mathbf{X}_n$ and the realizations $\mathbf{x}_1, \ldots, \mathbf{x}_n$. We define the log-likelihood as:

$$l(\beta) = \sum_{i=1}^{n}(y_i \log S(\beta^T\mathbf{x}_i) + (1 - y_i)(1 - \log S(\beta^T\mathbf{x}_i))).$$

This expression will be important later. In this particular case, we do not need to estimate β coefficient directly. We can change the variables

$$\beta_k = S^{-1}(\pi_k) = \log \frac{\pi_k}{1-\pi_k}.$$

Denote

$$a_{k1} = \sum_{i=1}^{n} 1_{\{x_i=k, y_i=1\}}, \quad a_{k\bullet} = \sum_{i=1}^{n} 1_{\{x_i=k\}}, \quad a_{k0} = a_{k\bullet} - a_{k1},$$

and the probability vector with $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)^T$. Then the log-likelihood function may be written as:

$$l(\boldsymbol{\pi}) = \sum_{k=1}^{K} (a_{k1} \log \pi_k + a_{k0} \log(1-\pi_k)).$$

which is the sum of the individual likelihoods. Using basic differentiation, we may express the optimal solution of the maximization directly. So we have that the optimal solution is:

$$\pi_k^* = \frac{a_{k1}}{a_{k\bullet}}.$$

Using the result, we can construct the estimator as:

$$\hat{\pi}_k = \frac{\sum_{i=1}^{n} Y_i \mathbf{1}_{\{X_i=k\}}}{\sum_{i=1}^{n} \mathbf{1}_{\{X_i=k\}}}.$$

This is the same result as before, just written in a different way. Deducing from above we have:

$$E(\hat{\pi}_k | X_i = k\,,\, i \in \{1, \ldots, n\}) = E_X(\hat{\pi}_k) = \pi_k$$

and

$$E_X((\hat{\pi})_k - \pi_k) = \mathrm{Var}_X(\hat{\pi_k}) = \pi_k(1-\pi_k).$$

Here we are using E_X and Var_X as shorter notations for conditioning on predictors. Now, when we have set up the logistic regression framework, we introduce a technique we will use to improve the MSE of $\boldsymbol{\pi}$.

3 Penalization

3.1 Overview

Penalization or regularization is a method, primely used for the constrained optimization problems which have found many applications in the machine learning. Let us return to the empirical risk minimization problem:

$$f^*_{n,\mathcal{S}} = \arg\min_{f\in\mathcal{S}} \frac{1}{n}\sum_{i=1}^{n} l(y_i, f(x_i)).$$

Solution of this problem dependens on the set $\mathcal{S}$, which is introduced to prevent overfitting. For example, let $\mathcal{S}$ be composed from L_2 bounded functions, i.e $\|f\|_{L_2} \leq 1$. Then we have a constrained problem

$$\min_{f\in L_2} \frac{1}{n}\sum_{i=1}^{n} l(y_i, f(x_i)) \quad \text{such that} \quad \|f\|_{L_2} \leq 1.$$

Instead of solving constrained optimization problem above, it is easier to control the norm of the function, by adding an additional summand called penalty. That is:

$$\min_{f\in L_2} \frac{1}{n}\sum_{i=1}^{n} l(y_i, f(x_i)) + \lambda\|f\|_{L_2}.$$

Therefore, instead of the explicit construction of the set $\mathcal{S}$, we are controlling some property of the argument function to prevent overfitting. Formally, every penalized optimization problem consists of the risk summand and the penalization summand, which is dependent on the parameter λ. Parameter λ is tuned by the user.

Example 3.1. Let $X = \mathbb{R}$ and $Y = \mathbb{R}$, so we have a regression problem. We want to construct a prediction function $f : \mathbb{R} \to \mathbb{R}$ which will be smooth. Smoothness in this case means without sharp jumps and peeks. We also want the function to be enough time differentiable. We need to find a measure which properly controls smoothness. Assume that the function has very sharp peek at some position. That can be interpreted in the following way: the function is highly concave on some narrow interval around the peek. So, the absolute value of the second derivative on the interval is high. On the other side, if the truncation is low, like in the linear case, convexity/concavity is also low. From the interpretation we can see that the suitable measure is the second

derivative function. Because of that, to control the smoothness, we introduce the 2-norm of the second derivative as our penalization. We formulate the learning problem as:

$$\min \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int_{\mathbb{R}} (f''(x))^2 dx.$$

This example is called a cubic spline regression. The solution is obtained by using numerical approximation of an unknown function.

3.2 Penalizations for the linear models

Even though linear models, for example logistic regression, are interpreted as simpler ones, they are not used for complex prediction tasks. Due to their simplicity, they usually do not overfit. However, overfitting can occur, so special techniques of penalizations are developed for linear models. The well-known are Lasso penalization and the ridge penalization. Besides overfitting prevention, the penalizations have also other positive benefits. We define lasso penalization:

$$P(\boldsymbol{\beta}) = \lambda \sum_{i=1}^{m} |\beta_i|.$$

Its main application is the feature selection process. Feature selection is the task where we have to choose predictors which are relevant for a model among many possible predictors which are offered to us. There exist a theoretical guarantee that the lasso penalization will quickly force some of the coefficients to reach 0, or to be very close to 0. This can be interpreted as if they are irrelevant for the prediction problem. More about lasso penalization can be found in the articles which have popularized the method [17] [15]. To understand the importance of the ridge regression, let us first see the linear regression model:

$$E(Y|X_1, \ldots, X_m) = \beta_1 X_1 + \ldots \beta_m X_m.$$

Determining unknown coefficients is done using least squares method. If we denote with $\mathbf{X}$ the data matrix, the estimator is of the form:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}.$$

According to [12], this estimator is an unbiased estimator with the least possible variance. What is happening when the data from the different features are highly correlated? Then the matrix $\mathbf{X}^T\mathbf{X}$ has the determinant close to 0, which leads to instability of the estimates. This phenomena is called bad conditioning or multicollinearity. Bad conditioning occurs very often in practice, so it is important to handle such exception.

In order to deal with it, we use the ridge penalization. We define the ridge penalization as:

$$P(\boldsymbol{\beta}) = \frac{\lambda}{2}\sum_{i=1}^{m}\beta_i^2.$$

If we add it to a linear regression model, using the least squares method, the obtained estimator is:

$$\hat{\boldsymbol{\beta}} = (\lambda I + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y},$$

where λ is the penalization coefficient. Now, the matrix which needs to be inverted has increased its conditioning number as a measure of singularity. It can be adjusted with the parameter λ. In this example with ridge penalization, we have a theoretical guarantee that it can solve the particular issue. On the other side, for the logistic regression we are not able the explicitly express linear coefficients. However, from the empirical results, it has been shown that also in the case of the logistic regression, ridge and lasso penalization can help in solving the same issue [11].

We have seen that penalizations may be very useful for solving the model related issues which may arise. From now on, we will try to see if penalizations may be helpful also for our problem: reducing the MSE of the combined estimation. The main idea will be to express the MSE as a function of λ, and manipulating with λ to try to achieve a smaller MSE. We will try to achieve that goal by using the ridge penalization.

3.3 New estimator

Now, let us construct a new estimator using the ridge penalization. We use the encoding of categorical data described in the previous chapter. With the logistic regression model and the ridge penalization, we obtain the likelihood:

$$l^P(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) + P(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i \log S(\beta^T\mathbf{x}_i) + (1-y_i)(1-\log S(\beta^T\mathbf{x}_i))) - \lambda\sum_{i=1}^{m}\beta_i^2.$$

Since it is a maximization problem, we are adding the penalization with minus. Again, using change of variables $\beta_k = \log\frac{\pi_k}{1-\pi_k}$, we have that

$$\begin{aligned} l^P(\boldsymbol{\pi}) &= \sum_{k=1}^{K}(a_{k1}\log\pi_k + a_{k0}\log(1-\pi_k)) - \frac{\lambda}{2}\sum_{k=1}^{K}\log^2\frac{\pi_k}{1-\pi_k} \\ &= \sum_{k=1}^{K}(a_{k1}\log\pi_k + a_{k0}\log(1-\pi_k) - \frac{\lambda}{2}\log^2\frac{\pi_k}{1-\pi_k}). \end{aligned}$$

The last expression can be separated with respect to π_k, so optimization of the complete expression is indeed an optimization of each summand separately. That is, we are

optimizing expressions

$$l_k^P(\pi_k) = a_{k1}\log\pi_k + a_{k0}\log(1-\pi_k) - \frac{\lambda}{2}\log^2\frac{\pi_k}{1-\pi_k}, \quad k \in \{1,\dots,K\}$$

separately. The first derivatives are:

$$l_k^{P'}(\pi_k) = \frac{a_{k1}}{\pi_k} - \frac{a_{k0}}{1-\pi_k} - \lambda\log\frac{\pi_k}{1-\pi_k}\frac{1}{\pi_k(1-\pi_k)}.$$

Solving the equation where the derivative is equal to zero is impossible explicitly. The method proposed in [14] suggests to do one step of the Newton method to find the zero of the derivative. In that procedure, as an initial value, we take $\pi_k^{init} = \frac{1}{2}$. So we would have that:

$$\pi_k^{\lambda} = \frac{1}{2} - \frac{l_k^P(\frac{1}{2})}{l_k^{P'}(\frac{1}{2})}.$$

After performing that step, the obtained estimator is:

$$\hat{\pi}_k^{\lambda} = \frac{a_{k1}+2\lambda}{a_{k\bullet}+4\lambda}. \tag{3.1}$$

We will denote the vector estimator as $\hat{\boldsymbol{\pi}}^{\lambda} = (\hat{\pi}_1^{\lambda},\dots,\hat{\pi}_K^{\lambda})$. Obviously, if we take $\lambda = 0$, we obtain the unbiased estimator $\pi_k = \frac{a_{k1}}{a_{k\bullet}}$, while for very large lambda, we have that:

$$\lim_{\lambda\to\infty}\frac{a_{k1}+2\lambda}{a_{k\bullet}+4\lambda} = \frac{1}{2}.$$

We have a parametric family of the estimators with parameter λ. We would like to show that the appropriate choice of the parameter λ can improve MSE of the vector $\hat{\boldsymbol{\pi}}^{\lambda}$. This means that it will be better than the MSE of the vector $\hat{\boldsymbol{\pi}}$. Blagus et al. showed in [14] that there exist parameter λ which satisfies the described condition. Here, we will move one step further, constructing a concrete example of such λ. However, that λ will dependen on the real values of the probabilities, and it cannot be used directly in the estimator function. $MSE(\hat{\boldsymbol{\pi}}^{\lambda}) = MSE(\lambda)$ denotes mean squared error of the expression (3.1). After simple calculations, we have that

$$MSE(\lambda) = \frac{1}{K}\sum_{k=1}^{K}\frac{4\lambda^2(1-2\pi_k)^2 + a_{k\bullet}\pi_k(1-\pi_k)}{(a_{k\bullet}+4\lambda)^2}.$$

Since we would like to minimize the expression, we calculate the derivative. Again, easily we obtain that:

$$MSE'(\lambda) = \frac{1}{K}\sum_{k=1}^{K}\frac{8a_{k\bullet}(\lambda(1-2\pi_k)^2 - \pi_k(1-\pi_k))}{(a_{k\bullet}+4\lambda)^3}.$$

To get the optimal MSE depending on λ, we need to solve the equation $MSE'(\lambda) = 0$. It is not possible to solve it explicitly for $K > 2$ (in that case we have a problem to

find roots of a polynomial of degree greater than 4). Let $K = 1$. Then we have the following:

$$\frac{8a_{1\bullet}(\lambda(1-2\pi_1)^2 - \pi_1(1-\pi_1))}{(a_{1\bullet}+4\lambda)^3} = 0 \implies \lambda_{opt} = \frac{\pi_1(1-\pi_1)}{(1-2\pi_1)^2}.$$

This solution exists when $\pi_1 \neq \frac{1}{2}$. The second special case represents the scenario when the sample size of all categories is the same, that is $a_{1\bullet} = \cdots = a_{K\bullet} = a_\bullet$. Then, we have the following:

$$\begin{aligned} MSE(\lambda) &= \frac{1}{K}\sum_{k=1}^{K} \frac{8a_{k\bullet}(\lambda(1-2\pi_k)^2 - \pi_k(1-\pi_k))}{(a_{k\bullet}+4\lambda)^3} \\ &= \frac{8a_\bullet}{K(a_\bullet+4\lambda)^3}\sum_{k=1}^{K}(\lambda(1-2\pi_k)^2 - \pi_k(1-\pi_k)) \\ &= \frac{8a_\bullet}{K(a_\bullet+4\lambda)^3}(\lambda\sum_{k=1}^{K}(1-2\pi_k)^2 - \sum_{k=1}^{K}\pi_k(1-\pi_k)) = 0 \\ &\implies \lambda_{opt} = \frac{\sum_{k=1}^{K}\pi_k(1-\pi_k)}{\sum_{k=1}^{K}(1-2\pi_k)^2}. \end{aligned} \tag{3.2}$$

These two special cases are appearing often in practice. Case $K = 1$ is estimation of the probability of success in a population. The case when all sample sizes are equal, may be interpreted as a case when we are sampling binary vectors of size K, where each coordinate of a vector has a Bernoulli distribution. The result exist only if all probabilities are not equal to $\frac{1}{2}$. In that case, the $MSE(\lambda)$ is a decreasing function, and optimality is reached at $\lambda \to \infty$. Like in the general case, we are not able to find the optimal solution explicitly. We will construct the suboptimal solution. We provide the following lemma.

Lemma 3.2. *Let g be two times continuously differentiable function on some domain A such that the second derivative is bounded on the domain, and we denote $L = \max_{x\in A}|g''(x)|$. Fix some $x \in A$ and let $y = x - \frac{1}{L}g'(x)$. If $y \in A$, then the following holds:*

$$g(y) - g(x) \leq -\frac{1}{2L}g'(x)^2.$$

Proof. From the Taylor theorem, there exist $\xi \in (0,1)$ such that

$$g(y) = g(x) + g'(x)(y-x) + \frac{1}{2}g''(x+\xi(y-x))(y-x)^2$$

Using the defined bound of the second derivative we have:

$$g(y) \leq g(x) + g'(x)(y-x) + \frac{1}{2}L(y-x)^2.$$

Replacing $y - x$ with $-\frac{1}{L}g'(x)$, we have that

$$g(y) - g(x) \leq -\frac{1}{L}g'(x)^2 + \frac{1}{2L}g'(x)^2 = -\frac{1}{2L}g'(x)^2.$$

□

Lemma 3.2 is showing us that when we have a function with bounded second derivative, then there is a way to construct a point where the value of the function is less or equal than the current value, with a certain gap. Since unbiased case is when $\lambda = 0$, our goal is to construct λ^* such that $MSE(\lambda^*) < MSE(0)$. For that purpose, we need to calculate and bound the second derivative of the MSE function on the positive axis. So we calculate the second derivative:

$$MSE''(\lambda) = \frac{1}{K}\sum_{k=1}^{K} \frac{8a_{k\bullet}((1-2\pi_k)^2(a_{k\bullet} - 8\lambda) + 12\pi_k(1-\pi_k))}{(a_{k\bullet} + 4\lambda)^4}.$$

Denote k-th summand with f_k. We introduce the following notations:

$$z_k = 8a_{k\bullet}^2(1-2\pi_k)^2 + 96a_{k\bullet}\pi_k(1-\pi_k),$$

$$b_k = 64a_{k\bullet}(1-2\pi_k)^2.$$

Then we can write

$$f_k(\lambda) = \frac{z_k - b_k\lambda}{(a_{k\bullet} + 4\lambda)^4}.$$

We construct the upper bound:

$$\begin{aligned}
|f_k(\lambda)| &= \left|\frac{z_k - b_k\lambda}{(a_{k\bullet} + 4\lambda)^4}\right| = \left|\frac{z_k - \frac{b_k}{4}4\lambda - \frac{a_{k\bullet}b_k}{4} + \frac{a_{k\bullet}b_k}{4}}{(a_{k\bullet} + 4\lambda)^4}\right| = \left|\frac{z_k + \frac{a_{k\bullet}b_k}{4}}{(a_{k\bullet} + 4\lambda)^4} - \frac{b_k}{4(a_{k\bullet} + 4\lambda)^4}\right| \\
&\leq \left|\frac{z_k + \frac{a_{k\bullet}b_k}{4}}{(a_{k\bullet} + 4\lambda)^4}\right| + \left|\frac{b_k}{4(a_{k\bullet} + 4\lambda)^3}\right| = \frac{z_k + \frac{a_{k\bullet}b_k}{4}}{(a_{k\bullet} + 4\lambda)^4} + \frac{b_k}{4(a_{k\bullet} + 4\lambda)^3} \\
&\leq \frac{z_k}{a_{k\bullet}^4} + \frac{a_{k\bullet}b_k}{4a_{k\bullet}^4} + \frac{b_k}{4a_{k\bullet}^3} = \frac{z_k}{a_{k\bullet}^4} + \frac{b_k}{2a_{k\bullet}^3}.
\end{aligned}$$

Using the inequality between arithmetic and geometric mean, for each k we have that:

$$\pi_k(1-\pi_k) \leq \left(\frac{\pi_k + 1 - \pi_k}{2}\right)^2 = \frac{1}{4}.$$

Using the last inequality, for the particular members we have that:

$$b_k \leq 64a_{k\bullet}, \quad z_k \leq 8a_{k\bullet}^2 + 24a_{k\bullet}.$$

Using the previous inequalities we finalize the calculation:

$$|f_k''(\lambda)| \leq \frac{8a_{k\bullet}^2 + 24a_{k\bullet}}{a_{k\bullet}^4} + \frac{64a_{k\bullet}}{2a_{k\bullet}^3} = \frac{40a_{k\bullet} + 24}{a_{k\bullet}^3}.$$

From the obtained upper bound for the individual summand, we further conclude that

$$|MSE''(\lambda)| \leq \frac{1}{K}\sum_{k=1}^{K}\frac{40a_{k\bullet}+24}{a_{k\bullet}^3}.$$

We know that for $\lambda = 0$, it holds that

$$MSE'(0) = -\frac{1}{K}\sum_{k=1}^{K}\frac{8\pi_k(1-\pi_k)}{a_{k\bullet}^2},$$

where the last expression is less than 0, except in the case when all probabilities are equal 0 or 1 in the same time. We have now prepared all prerequisites to state the following theorem.

Theorem 3.3. *Assume that not all probabilities are equal to* 0 *or* 1 *in the same time. Then, there exists* $\lambda^* > 0$ *such that* $MSE(\lambda^*) < MSE(0)$*, and it holds that:*

$$\lambda^* = \frac{\sum_{k=1}^{K}\frac{\pi_k(1-\pi_k)}{a_{k\bullet}^2}}{\sum_{k=1}^{K}\frac{5a_{k\bullet}+3}{a_{k\bullet}^3}}. \tag{3.3}$$

Proof. Using Lemma 3.2 we construct λ^* as:

$$\lambda^* = 0 - \frac{1}{L}MSE'(0) = -\frac{1}{L}MSE'(0), \tag{3.4}$$

where L is the upper bound of the second derivative of the MSE. We have shown that $|MSE''(\lambda)| \leq \frac{1}{K}\sum_{k=1}^{K}\frac{40a_{k\bullet}+24}{a_{k\bullet}^3}$, so it is suitable to take

$$L = \frac{1}{K}\sum_{k=1}^{K}\frac{40a_{k\bullet}+24}{a_{k\bullet}^3}$$

Putting calculated L and previously calculated value of the $MSE'(0)$ in the expression (3.4), we obtain (3.3). Due assumption that not all probabilities equal to 0 or 1, we have that $MSE'(0) < 0$. Because of that, we have

$$MSE(\lambda^*) - MSE(0) \leq -\frac{1}{2L}MSE'(0)^2 < 0 \quad \text{with} \quad \lambda^* > 0.$$

. □

So for the general case, we constructed the suboptimal value of λ. All constructed λ-s are directly dependent on the real probabilities which are unknown in the process of the estimation. Because of that, we need to estimate an optimal or suboptimal λ. In the future chapters, we will try to check the properties of the estimated λ with a simulation study.

In the theorem 3.3, we have the assumption that not all probabilities should be 0 or 1. If we allow that, we would have $\lambda^* = 0$. In that case, we do not overperform the unbiased estimator. The question is, can we overperform the unbiased estimator in every possible case. Without loss of generality, assume that all probabilities are 0. Then, since we are sampling from the discrete variable, we cannot have a sample point different than 0. So, with all zeros in the sample, we would have an estimator which is always zero. That estimator has a bias equal zero and variance is also 0. Therefore, the MSE in that case is zero meaning that we cannot overperform MSE. This observation leads us to the following lemma:

Lemma 3.4. *Assume that not all probabilities are equal to* 0 *or* 1 *in the same time. Then, there is no* $\lambda^* > 0$*, which satisfies* $MSE(\lambda^*) \leq MSE(0)$ *and such that it does not depend on* π_k*.*

Proof. Assume that there exist such $\lambda^* > 0$ independent from the probabilities π_k. We will construct a counter example. Take that $\pi_1 = \pi_2 = \cdots = \pi_k = \pi$. Then we have the function:

$$MSE(\lambda, \pi) = \frac{1}{K} \sum_{k=1}^{K} \frac{4(\lambda^*)^2(1-2\pi)^2 + a_{k\bullet}\pi(1-\pi)}{(a_{k\bullet} + 4\lambda^*)^2}.$$

Our goal is to find π such that $MSE(\lambda^*, \pi) > MSE(0, \pi)$. We define

$$f(\pi) = MSE(\lambda^*, \pi) - MSE(0, \pi).$$

Obviously, we have that such defined function is continuous in the second argument on the whole real line. Further we have:

$$f(0) = MSE(\lambda^*, 0) - MSE(0, 0) = \frac{1}{K} \sum_{k=1}^{K} \frac{4(\lambda^*)^2}{(a_{k\bullet} + 4\lambda^*)^2} > 0,$$

because $\lambda^* > 0$. Due to continuty, there exist ϵ such that $f(\epsilon) > 0$. That implies $MSE(\lambda^*, \epsilon) > MSE(0, \epsilon)$. Therefore, taking that $\pi_1 = \cdots = \pi_K = \epsilon$ is our counter example. □

We will prove another useful property.

Theorem 3.5. *Assume that not all probabilities are equal to* 0 *or* 1 *in the same time. Let* λ^* *be as in the Theorem 3.3. Then for each* λ^{**}, $0 < \lambda^{**} < \lambda^*$ *holds* $MSE(\lambda^{**}) < MSE(0)$*.*

Proof. By the construction of the λ^*, we know that there exist some L such that $\lambda^* = -\frac{1}{L} MSE'(0)$. Fix some λ^{**} between 0 and λ^*, and let

$$\epsilon = \frac{MSE'(0)}{\lambda^*} - \frac{MSE'(0)}{\lambda^{**}}.$$

Since $MSE'(0) < 0$, it holds that $\epsilon > 0$. Since L was the upper bound of the $|MSE''(\lambda)|$, so is also $L + \epsilon$. From the construction of ϵ we have that

$$\lambda^{**} = -\frac{1}{L+\epsilon} MSE'(0),$$

and obviously it holds that $\lambda^{**} > 0$. Because of that, λ^{**} satisfies also assumptions of the Lemma 3.2, so we have that

$$MSE(\lambda^{**}) - MSE(0) \leq -\frac{1}{2L} MSE'(0)^2 < 0.$$

□

3.4 Generalized Mean Squared Error

Sometimes, we may not be satisfied with the MSE risk in the case when we are estimating multiple parameters in the vector form. In the MSE, every parameter contributes uniformly to the final risk. On the other side, we may consider some parameters as more important to us. For those more important, we want to ensure that they will be estimated better than the others. Also, MSE does not take into account the mutal dependence between estimators. In order to take that into account, we may add a contributon of the correlations between the estimators to the final risk. Due to all these reasons, we generalize our MSE risk into Generalized Mean Squared Error (GMSE). For that purpose, we define generalized square loss as:

$$l(\boldsymbol{y}, \hat{\boldsymbol{y}}) = (\boldsymbol{y} - \hat{\boldsymbol{y}})^T \boldsymbol{B} (\boldsymbol{y} - \hat{\boldsymbol{y}}),$$

where $\boldsymbol{B}$ is a positive semidefinite matrix. The risk based on this loss is GMSE. Now we will prove a similar result for GMSE as the one in the previous section. We do that for an arbitrary positive semidefinite matrix $\boldsymbol{B}$. We exclude the trivial case when $\boldsymbol{B} = 0$. We have vectors $\hat{\boldsymbol{\pi}}^\lambda = (\hat{\pi}_1^\lambda, \ldots, \hat{\pi}_K^\lambda)$ and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$.We define GMSE as a function of λ:

$$GMSE(\lambda) = E((\hat{\boldsymbol{\pi}} - \boldsymbol{\pi})^T \boldsymbol{B} (\hat{\boldsymbol{\pi}} - \boldsymbol{\pi})).$$

Using simple algebraic calculations we have that:

$$GMSE(\lambda) = \boldsymbol{h}^T(\lambda) \boldsymbol{B} \boldsymbol{h}(\lambda) + \boldsymbol{g}^T(\lambda) \boldsymbol{D} \boldsymbol{g}(\lambda),$$

where $\boldsymbol{h} = (h_1, \ldots, h_K)^T$, $\boldsymbol{g}(\lambda) = (g_1, \ldots, g_K)^T$,

$$h_k(\lambda) = \frac{2\lambda(1 - 2p_k)}{a_{k\bullet} + 4\lambda}, \quad g_k(\lambda) = \frac{\sqrt{a_{k\bullet}\pi_k(1 - \pi_k)}}{a_{k\bullet} + 4\lambda}$$

and $\boldsymbol{D} = diag(\{B_{kk} \mid k \in \{1, \ldots, K\}\})$. We calculate derivatives using the chain rule:

$$GMSE'(\lambda) = \boldsymbol{h}^T(\lambda) \boldsymbol{B} \boldsymbol{h}'(\lambda) + \boldsymbol{h}^T(\lambda) \boldsymbol{B}^T \boldsymbol{h}'(\lambda) + 2\boldsymbol{g}^T(\lambda) \boldsymbol{D} \boldsymbol{g}'(\lambda),$$

$$GMSE''(\lambda) = \boldsymbol{h}^T(\lambda)\boldsymbol{B}\boldsymbol{h}''(\lambda) + \boldsymbol{h}'^T(\lambda)\boldsymbol{B}\boldsymbol{h}'(\lambda) + \boldsymbol{h}^T(\lambda)\boldsymbol{B}^T\boldsymbol{h}''(\lambda) + \boldsymbol{h}'^T(\lambda)\boldsymbol{B}^T\boldsymbol{h}'(\lambda) + 2\boldsymbol{g}^T(\lambda)\boldsymbol{D}\boldsymbol{g}''(\lambda) + 2\boldsymbol{g}'^T(\lambda)\boldsymbol{D}\boldsymbol{g}'(\lambda),$$

where the derivative of the vectors is the component-wise derivative. The previous expression is a sum of the members of the form x^TAy for some vectors x and y, and a positive semidefinite matrix A. We provide a bound for such forms:

$$x^TAy = \langle x, Ay\rangle \leq \|x\|_2\|Ay\|_2 \leq \|x\|_2\|A\|_2\|y\|_2.$$

The first inequality is the Cauchy Swartz one, while the second is coming from the properties of the matrix norms. For matrices A and B and any matrix norm, it holds that $\|AB\| \leq \|A\|\|B\|$. For the 2-norm of the matrix A, we know that it is the largest singular value of A. A singular value of A is a square root of an eigenvalue of A^TA. If A is a symmetric matrix, then the largest singular value is the largest eigenvalue. Denote that singular value with σ_{max}. We have that

$$x^TAy \leq \sigma_{max}\|x\|\|y\|.$$

Denote with b_{max} the largest singular value of $\boldsymbol{B}$ and with d_{max} the largest singular value of $\boldsymbol{D}$, which is the largest eigenvalue of $\boldsymbol{D}$. We know that d_{max} is the largest diagonal value of the matrix $\boldsymbol{B}$. Then we have that:

$$GMSE''(\lambda) \leq 2b_{max}(\|\boldsymbol{h}''(\lambda)\|\|\boldsymbol{h}(\lambda)\| + \|\boldsymbol{h}'(\lambda)\|^2) + 2d_{max}(\|\boldsymbol{g}''(\lambda)\|\|\boldsymbol{g}(\lambda)\| + \|\boldsymbol{g}'(\lambda)\|^2).$$

For the functions g and h, it holds the following:

$$|h_k(\lambda)| = \frac{2\lambda|1-2p_k|}{a_{k\bullet}+4\lambda} = \frac{|1-2\pi_k|}{2} - \frac{a_{k\bullet}|1-2\pi_k|}{2(a_{k\bullet}+4\lambda)} \leq \frac{|1-2\pi_k|}{2} \leq \frac{1}{2}$$

$$|h_k'(\lambda)| = \frac{2a_{k\bullet}|1-2\pi_k|}{(a_{k\bullet}+4\lambda)^2} \leq \frac{2a_{k\bullet}}{a_{k\bullet}^2} = \frac{2}{a_{k\bullet}}.$$

$$|h_k''(\lambda)| = \frac{16a_{k\bullet}|1-2\pi_k|}{(a_{k\bullet}+4\lambda)^3} \leq \frac{16a_{k\bullet}}{a_{k\bullet}^3} = \frac{16}{a_{k\bullet}^2}.$$

$$|g_k(\lambda)| = \frac{\sqrt{a_{k\bullet}\pi_k(1-\pi_k)}}{a_{k\bullet}+4\lambda} \leq \frac{\sqrt{a_{k\bullet}}}{2a_{k\bullet}}$$

$$|g_k'(\lambda)| = \frac{4\sqrt{a_{k\bullet}\pi_k(1-\pi_k)}}{(a_{k\bullet}+4\lambda)^2} \leq \frac{2\sqrt{a_{k\bullet}}}{a_{k\bullet}^2}$$

$$|g_k''(\lambda)| = \frac{32\sqrt{a_{k\bullet}\pi_k(1-\pi_k)}}{(a_{k\bullet}+4\lambda)^3} \leq \frac{16\sqrt{a_{k\bullet}}}{a_{k\bullet}^3}.$$

For the vectors we have the following bounds:

$$\|\boldsymbol{h}(\lambda)\| \leq \frac{\sqrt{K}}{2}, \quad \|\boldsymbol{h}'(\lambda)\| \leq 2\sqrt{\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^2}}, \quad \|\boldsymbol{h}''(\lambda)\| \leq 16\sqrt{\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^4}}$$

$$\|\boldsymbol{g}(\lambda)\| \leq \frac{1}{2}\sqrt{\sum_{k=1}^{K}\frac{1}{a_{k\bullet}}}, \quad \|\boldsymbol{g}'(\lambda)\| \leq 2\sqrt{\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^3}}, \quad \|\boldsymbol{g}'(\lambda)\| \leq 16\sqrt{\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^5}},$$

and finally, it holds that:

$$GMSE''(\lambda) \leq 2b_{max}\left(8\sqrt{K\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^4}}+4\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^2}\right)+2d_{max}\left(8\sqrt{\sum_{k=1}^{K}\frac{1}{a_{k\bullet}}}\sqrt{\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^5}}+4\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^3}\right).$$

To find a suitable λ, we need to calculate the derivative of the GMSE at the zero point. We have that:

$$\begin{aligned} GMSE'(0) &= 2\boldsymbol{h}^T(0)\boldsymbol{B}\boldsymbol{h}'(0) + 2\boldsymbol{g}^T(0)\boldsymbol{D}\boldsymbol{g}'(0) = 2\boldsymbol{g}^T(0)\boldsymbol{D}\boldsymbol{g}'(0) \\ &= 2\sum_{k=1}^{K} B_{kk}g_k(0)g_k'(0) = -2\sum_{k=1}^{K}\frac{4a_{k\bullet}B_{kk}\pi_k(1-\pi_k)}{a_{k\bullet}^3} \\ &= -8\sum_{k=1}^{K}\frac{B_{kk}\pi_k(1-\pi_k)}{a_{k\bullet}^2}. \end{aligned}$$

We provide a simple lemma.

Lemma 3.6. *For the nonzero positive semidefinite matrix A, all diagonal elements are nonnegative, with at least one strictly positive.*

Proof. Let e_i be the i-th vector of the standard base. Then, we have that $e_i^T A e_i \geq 0 \Leftrightarrow A_{ii} \geq 0$. So, all diagonal elements are nonnegative. Since the rank is greater than 0, we have that the sum of the eigenvalues is greater than 0. Since the sum of the eigenvalues is same as the trace, we have that the trace is also greater than 0. Given that all diagonal elements are nonnegative and that their sum is strictly positive, there exist at least one strictly positive diagonal element. □

From the Lemma 3.6 we have that all $B_{kk} \geq 0$ with at least one strictly greater than 0. Because of that holds $GMSE'(0) < 0$.

Theorem 3.7. *Assume that not all probabilities are equal to 0 or 1 in the same time. Then there exist λ^* such that $GMSE(\lambda^*) < GMSE(0)$, and it holds that*

$$\lambda^* = \frac{\sum_{k=1}^{K}\frac{B_{kk}\pi_k(1-\pi_k)}{a_{k\bullet}^2}}{b_{max}\left(2\sqrt{K\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^4}}+\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^2}\right)+d_{max}\left(2\sqrt{\sum_{k=1}^{K}\frac{1}{a_{k\bullet}}}\sqrt{\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^5}}+\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^3}\right)}.$$

Proof. Under the assumption that not all probabilities are equal 0 or 1, we have that $GMSE'(0) < 0$. On the other side, we have that the second derivative is bounded with the bound provided above. We denote it with L. Using the lemma we construct λ^* as

$$\lambda^* = 0 - \frac{1}{L}GMSE'(0) = -\frac{1}{L}GMSE'(0),$$

which is the expression obtained above. Obviously it holds that

$$GMSE(\lambda^*) - GMSE(0) \leq -\frac{1}{2L} GMSE'(0)^2 < 0 \quad \text{with} \quad \lambda^* > 0.$$

. □

3.5 Out-of-sample Mean Squared Error

Since one of the objectives of the thesis is to improve the result from the article [4], we should also take into account the prediction part. That means we should apply our model to unseen data. For that purpose, we will assume that we are given a new data $X_1^{new}, \ldots, X_n^{new}$, based on which we predict the variables $Y_1^{new}, \ldots, Y_n^{new}$. We know that the predictor is coming from K different classes. So, the particular number of sample points from each class we denote with $a_{1\bullet\bullet}, \ldots, a_{K\bullet\bullet}$. Using that, we can write the out-of-sample mean squared error as:

$$MSEO(\lambda) = E(\sum_{k=1}^{K} a_{k\bullet\bullet}(Y - \hat{\pi}_k^{\lambda})^2), \tag{3.5}$$

where Y from a different summand $(Y - \pi_k^{\lambda})^2$ is different, independent from the others, and is distributed $Y \sim Bern(\pi_k)$. For the particular summand, we have:

$$\begin{aligned} E((Y - \hat{\pi}_k^{\lambda})^2) &= E((Y - \pi_k + \pi_k - \hat{\pi}_k^{\lambda})^2) \\ &= E((Y - \pi_k)^2) + E((\pi_k - \hat{\pi}_k^{\lambda})^2) + 2E((Y - \pi_k)(\pi_k - \hat{\pi}_k^{\lambda})). \end{aligned}$$

The first term in the previous expression is the variance of the variable Y. The second therm is the MSE of the estimator $\hat{\pi}_k^{\lambda}$. The third one is 0, since $E(Y - \pi_k) = 0$ and due to independence. So we have that:

$$E((Y - \hat{\pi}_k^{\lambda})^2) = \pi_k(1 - \pi_k) + E((\pi_k - \hat{\pi}_k^{\lambda})^2).$$

Putting this back into (3.5), we have that:

$$MSEO(\lambda) = \sum_{k=1}^{K} a_{k\bullet\bullet}\pi_k(1 - \pi_k) + \sum_{k=1}^{K} a_{k\bullet\bullet}E((\pi_k - \hat{\pi}_k^{\lambda})^2.$$

The first term in the previous equation does not depend on λ, so we can ignore it for the optimization procedure. The second therm in the previous equation can be seen as the weighted MSE, where the weights are proportional to the test data size. Further, we can see it as the GMSE where our $K \times K$ positive semidefinite matrix is

$diag(a_{1\bullet\bullet}, \ldots, a_{K\bullet\bullet})$. So, to find some λ for which we have a smaller $MSEO$, we can use the Theorem 3.7, where we have:

$$\lambda^* = \frac{\sum_{k=1}^{K} \frac{a_{k\bullet\bullet}\pi_k(1-\pi_k)}{a_{k\bullet}^2}}{a_{\bullet\bullet}\left(2\sqrt{K\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^4}} + \sum_{k=1}^{K}\frac{1}{a_{k\bullet}^2} + 2\sqrt{\sum_{k=1}^{K}\frac{1}{a_{k\bullet}}}\sqrt{\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^5}} + \sum_{k=1}^{K}\frac{1}{a_{k\bullet}^3}\right)},$$

where $a_{\bullet\bullet} = \max_{k\in\{1,\ldots,K\}}\{a_{k\bullet\bullet}\}$. This out of sample case will have more meaning in the simulation study. It has practical motivation in data from the Brown's article.

3.6 Other estimators

In this section we present other possible estimators which can be constructed using ridge regression. Namely, the encoding which we provided for categorical data is not unique. The previous idea was that we have a binary vector $\mathbf{X} = (X_1, \ldots, X_k)$ which was the numerical equivalent of the categorical variable X. For the vector $\mathbf{X}$, we have numerical data which lies in euclidean space. We known that for a numerical sample it holds that if we perform a bijective linear transformation on each sample point, we do not loose any information. So, we can linearly transform the encoded vector $\mathbf{X}$ with a non-singular matrix $\mathbf{A}$. The new encoding will be $\mathbf{AX}$. For already described encoding $\mathbf{X}$, the matrix A is the identity matrix. The other well-known encoding is when one row of that identity matrix is replaced with the vector of ones. Let us take the last row of the matrix $\mathbf{A}$ to be the row of ones. Then another encoding we obtain is $\mathbf{X}^{(2)} = (X_1, \ldots, X_{k-1}, X_1 + \cdots + X_k)$. By the construction of the X_i, we have that $X_1 + \cdots + X_k = 1$.

So, instead of having a vector $\mathbf{X} = (X_1, \ldots, X_k)$ in the learning process, we have $\mathbf{X}^{(2)} = (X_1, \ldots, X_{k-1}, 1)$, where the description of the X_i stays the same as before. So, we have a new penalized model:

$$l^P(\boldsymbol{\gamma}) = l(\boldsymbol{\gamma}) + P(\boldsymbol{\gamma}) = \sum_{i=1}^{n}(y_i \log S(\gamma^T \mathbf{x}_i^{(2)}) + (1-y_i)(1 - \log S(\gamma^T \mathbf{x}_i^{(2)}))) - \lambda \sum_{i=1}^{m} \gamma_i^2.$$

Due to the fact that different encodings represent the same sample, we have that:

$$\boldsymbol{\beta}\mathbf{X} = \boldsymbol{\gamma}\mathbf{X}^{(2)}. \tag{3.6}$$

From that expression, we express new coefficients using old ones:

$$\gamma_K = \beta_K \quad \text{and} \quad \gamma_k = \beta_k - \beta_K.$$

Using already performed change of variables $\beta_k = \log \frac{\pi_k}{1-\pi_k}$, we have that

$$\gamma_k = \log \frac{\pi_k}{1-\pi_k} - \log \frac{\pi_K}{1-\pi_K} = \log \frac{\pi_k(1-\pi_K)}{\pi_K(1-\pi_k)} \quad \text{for} \quad k \in \{1, \ldots K-1\}$$

and $\gamma_K = \log \frac{\pi_K}{1-\pi_K}$. We apply the penalization to all parameters except the intercept. Using that, together with (3.6), we reformulate our model as:

$$\begin{aligned} l^P(\boldsymbol{\pi}) &= \sum_{k=1}^{K}(a_{k1}\log\pi_k + a_{k0}\log(1-\pi_k)) - \frac{\lambda}{2}\sum_{k=1}^{K-1}\log^2\frac{\pi_k(1-\pi_K)}{\pi_K(1-\pi_k)} \\ &= \sum_{k=1}^{K-1}(a_{K1}\log\pi_k + a_{K0}\log(1-\pi_k) - \frac{\lambda}{2}\log^2\frac{\pi_k(1-\pi_K)}{\pi_K(1-\pi_k)}) \\ &+ (a_{K1}\log\pi_K + a_{K0}\log(1-\pi_K). \end{aligned}$$

So again, every summand is dependent on only one π_k, so we can optimize each summand separately. Using the same approach as above, Blagus et al. showed that [14]:

$$\hat{\pi}_K^{\lambda,2} = \frac{a_{K1} + 4\lambda\sum_{i=1}^{K-1}\frac{a_{j1}}{a_{j\bullet}+4\lambda}}{a_{K\bullet} + 4\lambda\sum_{j=1}^{K-1}\frac{a_{j\bullet}}{a_{j\bullet}+4\lambda}} \quad \text{and} \quad \hat{\pi}_k^{(2)} = \frac{a_{k1} + 4\lambda\hat{\pi}_K^{\lambda,2}}{a_{k1\bullet}+4\lambda},\ k \in \{1,\ldots,K-1\}.$$

In some way, we can try with different encodings, despite the fact that such encodings do not have any meaning in practice. For example we can define a new encoding as $\mathbf{X}^{(3)} = (X_1,\ldots,X_k,1)$ which we will call overparametrizing. It leads to a singular data matrix $\underline{\mathbf{X}}$. Optimization in that case would not be possible without a penalization. Using penalization we create interesting estimators. This approach is shown in [14], and after similar calculations as before, we obtain the estimator

$$\hat{\pi}_k^{\lambda,3} = \frac{a_{k1} + 4\lambda\bar{\pi}^{\lambda}}{a_{k1\bullet}+4\lambda},\ k \in \{1,\ldots,K-1\},$$

where:

$$\bar{\pi}^{\lambda} = \frac{\sum_{j=1}^{K}\frac{a_{j1}}{a_{j\bullet}+4\lambda}}{\sum_{j=1}^{K}\frac{a_{j\bullet}}{a_{j\bullet}+4\lambda}}.$$

Analyzing these estimators in the similar manner as before is tiring, so we leave it for an ambitious reader. We will test the properties of those estimators in simulation studies, using different methods for estimating λ.

4 Cross Validation

In this chapter we present the cross validation approach, its use in machine learning and how it can be used in our case. Cross validation is the most common approach used to evaluate the quality of a predictor in the supervised learning problem. Another important use of cross validation is to determine the parameter λ in penalized models, for example in ridge regression. The main idea of cross validation is a multiple division of a data set into the train and test set, described in Section 2.4. After the divisions are performed, we average results of the predictions on the test sets, to obtain the final score of our algorithm. Cross validation is also used in the model selection, since its quality evaluation is used to compare different models and to choose the best. Also cross validation may indicate a sort of overfitting if the variance of the averaging values is large. Here, we introduce a few cross validation methods and their positive and negative properties.

- **Monte Carlo cross validation**

  Monte Carlo cross validation is the most random cross validation method. We randomly divide our sample set into the train and test set, decent number of times. For each division, the model is fitted on the train set and evaluated on the test set. At the end, we average the evaluations of the test sets, and we obtain a final quality estimation. An advantage of this approach is that we can have a large number of random splits. That is because averaging a bigger number of different performances may reduce the variance of the final estimation of the quality. However, a disadvantage is that we are randomly splitting the dataset which can cause that the train and test sets overlap among different splits (they will have common elements). That will induce a covariance between estimated predictors and variables from the test set. That can lead to a huge variance of the final quality estimation. The size ratio of the train and test set is described in Section 2.4.

- **k-fold cross validation**

  In this approach we randomly divide our sample set S into k approximately equally sized folds: $S_1, \ldots, S_k$. We train k different predictors on the train sets $S \setminus S_k$, and evaluate them on the test sets S_k. A positive side of this method is

that we use each point for the evaluation only once, so there are no overlapping between the different test sets. This further implies that we got rid of the variance induced by the overlapping of the test sets. If we follow the advice from Section 2.4, then the number of predictors will be small (between 5 and 10). Therefore, a disadvantage is that we will do averaging with a small number of predicting values. This again leads to a larger variance. Number k is in practice usually 5 or 10, again because of the reasons described in Section 2.4.

- **leave-one out cross validation (LOOCV)**

  This method can be seen as a special case of the previous one, when it holds that $k = |S|$. That means we are leaving only one element as the evaluation part. The positive side in comparison to the k-fold cross validation with bigger k is that we are using more complete set to train our model. That will reduce the bias (assuming model does not overfitt) since we are using more data to train the model. On the other side, only one sample point is left out which can increase both bias and variance in evaluation of the test set. That is possible because we may have a non-representative sample of the data distribution, i. e. outlier. We reduce the increment of the variance with a large number of splits (which is equal to $|S|$). A practical advantage of this method, compared to all other cross validation methods, is that it is completely non-random, and the final evaluation of the model can be expressed as a deterministic function. We will use this property to determine the unknown parameter λ, such that we maximize the performance of the model. The negative side of this approach is that it can be computationally demanding. The number of models which we need to train is $|S|$. For a large sample size $|S|$ and some complex model, it can be time consuming.

4.1 Determining λ

In the previous chapter, we have introduced the penalization approach. We have seen that it depends on the penalization coefficient λ, for which we said that it is tuned by the user. Often, the right value which should be tuned is not so obvious. Therefore, we need to construct a method which can be used to determine it more precisely. Now the question is, why λ is not a part of the optimization process as an argument in the model training. In the objective function, λ is a part of the summand where it is multiplied with something positive. So, the optimization result will set λ to be 0. That will nullify the effects of the penalization, which further may lead to overfitting. Because of that, for the training procedure, we need to fix lambda, and then to optimize over other parameters in the objective. After that, we need to construct a secondary

objective function, based on the estimated predictor which is directly dependent on λ. The optimization of the secondary objective over λ will give us the best predictor. Of course, this is an idealistic concept, and direct optimization over λ is usually not possible in practice. Often, we construct some finite set of values which λ can take. Then we train different models with λ taking values from the defined set of values. We choose λ for which the defined secondary objective function is the best.

The only unknown thing now is the secondary objective function, and how to construct it. The problem of prediction is to define a predictor which will explain, in the best way, the unknown probability distribution. Therefore, we would like to have a correct result for any sample from that unknown distribution. That includes all values that have not been used for the training (for the terminology purpose, the data not used for training are called unknown data or unknown sample). Since we do not know the true distribution, we cannot sample from it. The only way to have some sample, which is unknown for the estimated predictor, is to keep some part of the existing data out of training. Using that part, we can evaluate our predictor. For that purpose, we would like that the secondary objective function in the best possible way describes the evaluation of the predictor on an unknown data set. For that task, one possibility is to use cross validation. The secondary objective function will be exactly the quality estimation obtained during cross validation. The reason for that lies in the explanation provided above. We want to have a score which is related to the evaluation on the unknown data set. The most used cross validation strategy for this task in practice is the k-fold cross validation. That is because it gives the best results. Moreover, it has the least computational complexity; we need to train only k models, which for $k = 5$ or $k = 10$ is affordable in practice. Now, we return to our idealistic concept for finding the optimal λ. In the k-fold cross validation, the secondary objective function is random. Optimization over a random function is not possible. Here, we can resort to LOOCV since it does not have randomness, and we will use exactly this approach to determine λ. In more details, for different loss functions we will construct a deterministic secondary objective function based on LOOCV. After that, we will directly optimize the function to get an estimation for the parameter λ. Let us now return to our estimator. We had that probability π_k is estimated using Ridge regression as:

$$\hat{\pi}_k = \frac{a_{k1} + 2\lambda}{a_{k\bullet} + 4\lambda}.$$

By the assumption, our sample consists of K folds coming from different Bernoulli distributions. Assume that we omitted one 0 from the k-th fold of our sample for $k \in \{1, \ldots, K\}$. Then by training logistic regression on the remaining data, we get the estimator for the k-th probability as:

$$\hat{\pi}_{k0} = \frac{a_{k1} + 2\lambda}{a_{k\bullet} - 1 + 4\lambda},$$

while the estimators for other folds are as in the case when we are using the full dataset. If we omit one 1 from the k-th fold, then our estimator will be

$$\hat{\pi}_{k1} = \frac{a_{k1} - 1 + 2\lambda}{a_{k\bullet} - 1 + 4\lambda},$$

while the others will remain the same. Now we need to evaluate the performance of the obtained estimator on the data point we omitted. To do that, we need to define a loss function, which will do the evaluation. For now, we will take the same loss that was used for the logistic regression - log loss mentioned in (2.3). In the case when 0 is omitted, we have that:

$$l(0, \hat{\pi}_{k0}) = -0 \log \hat{\pi}_{k0} + (1 - 0) \log(1 - \hat{\pi}_{k0}) = \log(1 - \hat{\pi}_{k0}),$$

while for the other case when 1 is omitted we have:

$$l(1, \hat{\pi}_{k1}) = 1 \log \hat{\pi}_{k1} + (1 - 1) \log(1 - \hat{\pi}_{k1}) = \log(\hat{\pi}_{k1}).$$

Now, after the evaluation at every point, we have that the cross validation is

$$D(\lambda) = -\sum_{k=1}^{K} (a_{k1} \log \hat{\pi}_{k1} + a_{k0} \log(1 - \hat{\pi}_{k0})).$$

Function $D(\lambda)$ is somewhere refereed as LOOCV deviance, and it will play a roll of a secondary objective function described above. Here we just summed without averaging, because average factor is not relevant for optimization. We have to notice that the deviance is not defined for the case when there is a sample fold which size is equal 1. That is because we did not train any model using that fold. The next loss function, with which we will try to improve our λ is the mean squared error. So, we have that:

$$l(0, \hat{\pi}_{k0}) = (0 - \hat{\pi}_{k0})^2 = \hat{\pi}_{k0}^2, \quad l(1, \hat{\pi}_{k1}) = (1 - \hat{\pi}_{k1})^2,$$

and at the end, as cross validation we have:

$$PE(\lambda) = \sum_{k=1}^{K} (a_{k1}(1 - \hat{\pi}_{k1})^2 + a_{k0}\hat{\pi}_{k0}^2).$$

We will make a small comparison with the previous chapter. In Chapter 3 we have directly computed the mean squared error of our estimator, which is a function of λ, but it is also a function of the real values of the unknown probabilities. We have seen that it is not possible to optimize λ directly, that is to express the optimal λ as a function of those probabilities. We provided one suboptimal solution which was dependent on unknown probabilities. We will estimate it and test its performance in the simulation studies. Given that we have unknown probabilities in the MSE function, we are not able

to perform any numerical optimization method. Therefore, from that part, we need to be satisfied with sub-optimality. In this chapter, we created new objective functions, based on the experimentally justified method - cross validation. These objectives are not dependent on real probabilities and for them in practice, we can perform numerical optimization methods to obtain λ. But, since we are not optimizing directly the MSE of the estimators, we do not know will the numerically obtained optimal λ improve MSE in any sense. We will try to get the answers in a simulation study. Since we will use numerical optimization methods, we provide a review of them.

4.2 Numerical optimization methods

Discarding some trivial and degenerate cases, we have a problem of optimizing a differentiable function. Since the function is continuous, we usually call this continuous optimization. In comparison with discrete optimization, our search space of all possible solutions is uncountable. Without loss of generality, we will assume that we have a minimization problem. Theory of continious optimization is usually divided into two branches, convex and non-convex optimization. Convex optimization is the one where the search domain is a convex set and where the objective function is convex. An advantage of this case is that we known that there exists only one optimal solution which is global. Therefore, first time when we detect some local minimum, we known that it is also a global minimum. For convex optimization, a lot of research has been done and there are many reliable methods for which we have a theoretical guarantee that they will lead us to an optimal solution. On the other side, in the non-convex optimization we are working with non-convex functions and there is a possibility that we have a lot of local minimum, but to find a global one is hard. A decent research is done also in this topic, but there is no guarantee that we can find a global minimum. At the moment, we need to be satisfied with heuristic and meta-heuristic solutions. However, here we will focus on the convex optimization approach. As we will see, both of our objective functions are non-convex. We will search for an interval where the function is convex, and using convex optimization methods, we will try to find a local minimum.

4.2.1 Gradient descent

First we will focus on problems without any constraints. Assume that we have a convex function $f(\boldsymbol{x})$ over a convex domain $A \in \mathbb{R}^n$ for some n. Our goal is to find a point $\boldsymbol{x}_0$ such that $f(\boldsymbol{x}_0) \leq f(\boldsymbol{x})$ for every $\boldsymbol{x} \in A$. We will give an intuition of the proposed methods. Imagine a convex function as some valley. We are somewhere on

the sides of the valley and we want to reach the bottom of it. Logically, we will move down all the time, and since there is only one bottom, we will reach it at one point. Using mathematical expressions, we start from some point of the function, and we are iterating such that the next member of the sequence, evaluated in f, will be smaller than the previous one. Formally written, we want to create a sequence $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots$ such that $f(\boldsymbol{x}_{i+1}) \leq f(\boldsymbol{x}_i)$ for every $i \in \mathbb{N}$. Since the function is bounded from below, the sequence will converge. We want to ensure that the function will not converge to some point before. From Calculus we know that the gradient is a vector which shows the direction of the greatest rate of increase of the function. Because of that, the minus of the gradient is showing us the direction of the greatest rate of decrease. So the gradient will help us with the direction, such that we will always go down. We formulate our iterating method as

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \alpha \nabla f(\boldsymbol{x}_n), \tag{4.1}$$

where α is a positive real parameter which is tuned by the user and operator ∇ stands for a gradient. Using this method on each step we choose the direction based on the gradient, and we move with a step proportional to α. To show that with this approach we can achieve a decreasing sequence, we help ourselves with the following lemma.

Lemma 4.1. *Assume that f is continuously differentiable function on the convex domain, and let $\boldsymbol{x}$ be a point from that domain where the gradient is not 0. Then, there exist ϵ such that*

$$f(\boldsymbol{x} - \epsilon \nabla f(\boldsymbol{x})) \leq f(\boldsymbol{x}).$$

Proof. We use the Taylor expansion of the first order. We have that:

$$f(\boldsymbol{x} - \epsilon \nabla f(\boldsymbol{x})) = f(\boldsymbol{x}) - \epsilon \langle \nabla f(\boldsymbol{x}), \nabla f(\boldsymbol{x} - \xi \epsilon \nabla f(\boldsymbol{x})) \rangle$$

for some $\xi \in (0, 1)$ and continuously dependent on ϵ. Denote with $g(\epsilon) = \langle \nabla f(\boldsymbol{x}), \nabla f(\boldsymbol{x} - \xi \epsilon \nabla f(\boldsymbol{x})) \rangle$. We have that $g(0) = \|f(\boldsymbol{x})\|^2 > 0$. Since g is continuous, there exist $\epsilon > 0$ such that $g(\epsilon) > 0$. Taking that ϵ we have that:

$$f(\boldsymbol{x} - \epsilon \nabla f(\boldsymbol{x})) = f(\boldsymbol{x}) - \epsilon g(\epsilon) < f(\boldsymbol{x}).$$

□

This lemma is showing us that on each step we can adjust α such that we have a decreasing sequence. By the construction, the sequence cannot converge somewhere before the minimum, since the derivative is different than 0. So, the question is how can we choose α, and should we choose a different α at the each iteration step. We provide an additional definition.

Definition 4.2. A differentiable function f is L-smooth on the domain A if its gradient is Lipschitz continuous with the coefficient L:

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|,$$

for all $\boldsymbol{x}, \boldsymbol{y} \in A$.

If in the previous definition, we take a limit $\boldsymbol{y} \to \boldsymbol{x}$, and do a bit of simple analysis, we obtain that $\nabla^2 f(\boldsymbol{x}) \leq L$, which means that the norm of the Hessian is bounded with L. This further means that all eigenvalues of the Hessian are less than L. So usually, to obtain the constant L in practice, we compute the largest eigenvalue of the Hessian by absolute value. The importance of the defined constant L is given in the following theorem.

Theorem 4.3. *Let f be a convex and L-smooth function defined on the convex domain. Let x_0 be a starting point. For the iteration process defined as:*

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \frac{1}{L}\nabla f(\boldsymbol{x}_n), \tag{4.2}$$

we have that:

$$f(\boldsymbol{x}_n) - f(\boldsymbol{x}^*) \leq \frac{2L\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}{n-1} = O\left(\frac{1}{n}\right).$$

Proof can be found in [10]. This theorem is telling us that if we choose that $\alpha = \frac{1}{L}$, we have a guarantee that the iteration will converge and that the order of the convergence is $O(\frac{1}{n})$. To improve the result, we define the strong convexity.

Definition 4.4. We say that f is a strongly convex function on the convex domain A, with the coefficient μ, if we have that

$$\langle \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \geq \mu\|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

If we assume that f is twice differentiable, the previus theorem implies that $\|\nabla^2 f(\boldsymbol{x})\| \geq \mu$. In this case μ is something opposite than L; while L was the upper bound of the Hessian, the μ is the lower bound of it. In one-dimensional euclidean space, this may be interpreted as if the second derivative is always strictly positive. With this we can formulate the following theorem.

Theorem 4.5. *Let f be a strongly convex function with coefficient μ, and L-smooth function defined on the convex domain. Let x_0 be the starting point, and let $0 < \alpha \leq \frac{1}{L}$. For the iteration process defined as:*

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \alpha\nabla f(\boldsymbol{x}_n), \tag{4.3}$$

we have that:

$$\|\boldsymbol{x}_n - \boldsymbol{x}^*\|^2 \leq (1 - \alpha\mu)^n\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2.$$

With the previous theorem we improved our convergence rate to the linear one. The smaller α is we have slower convergence. Therefore, the best rate we obtain when we take $\alpha = \frac{1}{L}$.

So we formulated a numerical method for the convex optimization. It has a theoretical guarantee that it will converge, regardless where we start the walk. As a disadvantage of the method, we have that it is not robust if we have points where the function is not differentiable. That is because we need a gradient defined in every point. Another disadvantage is that it is not suitable for the problems with constraints. In the next subsection, we will present some modifications of the gradient descent to solve these issues. Also, we will present methods which improve the speed of the convergence in practice.

4.2.2 Other gradient descent based methods

Here we will give just a brief overview of gradient descent based methods.

Proximal gradient descent

Imagine that we have a logistic regression model, and we want to do a feature selection. As we have already mentioned, LASSO penalization can be suitable for that, so we can formulate our problem as an optimization problem; to optimize the log-likelihood together with the lasso penalization. Since the LASSO penalization includes the absolute values of the coefficients, it is not differentiable at $\mathbf{0}$. So, the optimization in that case may be problematic. For that purpose, there exists a theory of proximal gradient descent which is used for objectives of the form $f(x) + g(x)$, where f is convex and differentiable, while g is not differentiable but "proximal friendly", which means that we can explicitly define an proximal operator:

$$\text{prox}_g(\boldsymbol{v}) = \arg\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{v}\|^2 + g(\boldsymbol{w}).$$

When we have defined such operator, we can define a proximal gradient descent as:

$$\boldsymbol{x}_{n+1} = \text{prox}_g(\boldsymbol{x}_n - \alpha \nabla f(\boldsymbol{x}_n)).$$

This approach can be used also for the constrained problem. If we have the objective function f and a constraint such that $\boldsymbol{x} \in A$ for some domain A, then we can define a function g; $g(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in A$ and $g(\boldsymbol{x}) = \infty$ otherwise. With such defined function g, we can redefine our objective as $f + g$, without constraints. For that case, we can show easily that the proximal operator is an orthogonal projection on the set A. We have iteration process defined as

$$\boldsymbol{x}_{n+1} = \text{proj}_A(\boldsymbol{x}_n - \alpha \nabla f(\boldsymbol{x}_n)),$$

where proj_A is a projection operator. Of course, constraints can be defined in many ways, and often it is impossible to define the set A where we need to project. That implies that we cannot define a projection neither. More about proximal gradient approach may be found in [5].

Quasi Newtonian methods

The reason why we develop those numerical methods is because we cannot solve the system of the equations $\nabla f(\boldsymbol{x}) = 0$. If we look from the other prespective, we can use the Newton method to solve such problem, and the iteration step will be defined as:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - [\nabla^2 f(\boldsymbol{x}_n)]^{-1} \nabla f(\boldsymbol{x}_n).$$

Calculating the Hessian matrix and then inverting it can be very computationally demanding task, so we would like to get rid of it. For that purpose, instead of calculating the inverse of the Hessian, we use its approximation. That approximation we improve at every iteration step, which at the end converges to the inverse of the Hessian. Still such methods are computationally demanding, but since the Newton method itself has quadratic convergence, this approximation method has the same. This is a better result compared to the linear convergence achieved with the classical gradient descent. More details about Quasi-Newton methods can be found in the PhD thesis [8].

5 Simulation study and final results

5.1 Estimating λ

A simulation study or empirical study in general is an approach we use when we are not able to theoretically prove a property of some estimator or some predictor. It consists usually of creating artificial data from a known distribution, and then applying the estimator which we created to those data in order to test estimator's properties. In our case we would like to test different estimators we obtained from the logistic regression with ridge penalization. They will be tested together with different estimates of λ. Our goal is to try to improve the MSE of the unbiased estimator. The reason why do we do simulation study is that we were not able to evaluate the MSE directly for different λ estimates.

For now, we have discussed some ways to obtain λ, but we did not express any concrete estimator of λ. From (3.3) we have that:

$$\lambda^* = \frac{\sum_{k=1}^{K} \frac{\pi_k(1-\pi_k)}{a_{k\bullet}^2}}{\sum_{k=1}^{K} \frac{5a_{k\bullet}+3}{a_{k\bullet}^3}}. \tag{5.1}$$

This expression depends on π_k which is unknown in practice. Therefore, we will estimate it by estimating the expression $\pi_k(1-\pi_k)$. One approach is to replace π_k with the unbiased estimator $\hat{\pi}_k$. So, we have the first λ estimator as:

$$\hat{\lambda}_1^* = \frac{\sum_{k=1}^{K} \frac{\hat{\pi_k}(1-\hat{\pi}_k)}{a_{k\bullet}^2}}{\sum_{k=1}^{K} \frac{5a_{k\bullet}+3}{a_{k\bullet}^3}}.$$

On the other side, we know that $\pi_k(1-\pi_k)$ is the variance of the Bernoulli distribution. Hence, we will also try with the unbiased estimator of that variance. We know from before that for a sample $X_1, \ldots, X_n$, with the unbiased mean estimator $\bar{X}$, the unbiased estimator of their varience is:

$$\hat{\sigma} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \hat{X})^2 = \frac{1}{n-1}(\sum_{i=1}^{n} X_i^2 - n\bar{X}^2).$$

Using notations from our problem, we have the estimator:

$$\hat{\sigma} = \frac{a_{k1}}{a_{k\bullet}-1} - \frac{a_{k1}^2}{a_{k\bullet}(a_{k\bullet}-1)}.$$

Therefore, the second approach gives the following estimator for λ:

$$\hat{\lambda}_2^* = \frac{\sum_{k=1}^{K}\left(\frac{a_{k1}}{a_{k\bullet}^2(a_{k\bullet}-1)} - \frac{a_{k1}^2}{a_{k\bullet}^3(a_{k\bullet}-1)}\right)}{\sum_{k=1}^{K}\frac{5a_{k\bullet}+3}{a_{k\bullet}^3}}.$$

In the Chapter 4, we have constructed two LOOCV objective functions, which minimization leads to a reasonable estimation of λ. Again, an explicit optimization is not possible due to complexity of the expression. So, we will construct an iterative procedure for the optimization. First, we will do it for the LOOCV deviance. We had that:

$$D(\lambda) = -\sum_{k=1}^{K}(a_{k1}\log\hat{\pi}_{k1} + a_{k0}\log(1-\hat{\pi}_{k0})).$$

The first derivative of the above expression is:

$$D'(\lambda) = -\sum_{k=1}^{K}\left(\frac{2a_{k1}(a_{k\bullet}-2a_{k1}+1)}{(a_{k1}-1+2\lambda)(a_{k\bullet}-1+4\lambda)} + \frac{2a_{k0}(a_{k\bullet}-2a_{k0}+1)}{(a_{k0}-1+2\lambda)(a_{k\bullet}-1+4\lambda)}\right).$$

After some calculations we can obtain that:

$$D'(0) = -\frac{2(a_{k0}^2+a_{k1}^2)-2a_{k\bullet}}{(a_{k1}-1)(a_{k0}-1)(a_{k\bullet}-1)}.$$

Using the inequality between means, it holds that $2(a_{k0}^2+a_{k1}^2) \geq (a_{k0}+a_{k1})^2 = a_{k\bullet}^2$. It follows that $2(a_{k0}^2+a_{k1}^2) > 2a_{k\bullet}$, under the assumption that $a_{k\bullet} > 2$. When this holds, we have that $D'(0) < 0$. With this, we ensure that the objective is decreasing at 0 That implies there is a local minimum on positive line (which can be at the infinity). The second derivative of $D(\lambda)$ is:

$$D''(\lambda) = \sum_{k=1}^{K}\left(\frac{4a_{k1}(a_{k\bullet}-2a_{k1}+1)(2a_{k1}+a_{k\bullet}-3+8\lambda)}{(a_{k1}-1+2\lambda)^2(a_{k\bullet}-1+4\lambda)^2} + \frac{4a_{k0}(a_{k\bullet}-2a_{k0}+1)(2a_{k0}+a_{k\bullet}-3+8\lambda)}{(a_{k0}-1+2\lambda)^2(a_{k\bullet}-1+4\lambda)^2}\right).$$

To perform the gradient descent algorithm, and to be sure that we will achieve the convergence, we need to find the smoothness constant. Since it was interpreted as the bound of the Hessian matrix in high-dimensional case, in one-dimensional case it can be interpreted as the bound of the second derivative. Therefore, we need to bound the

calculated second derivative. We have that

$$\begin{aligned}|D''(\lambda)| &= \Big|\sum_{k=1}^{K}\Big(\frac{8a_{k1}(a_{k\bullet}-2a_{k1}+1)}{(a_{k1}-1+2\lambda)(a_{k\bullet}-1+4\lambda)^2}+\frac{4a_{k1}(a_{k\bullet}-2a_{k1}+1)}{(a_{k1}-1+2\lambda)^2(a_{k\bullet}-1+4\lambda)}\\ &+\frac{8a_{k0}(a_{k\bullet}-2a_{k0}+1)}{(a_{k0}-1+2\lambda)(a_{k\bullet}-1+4\lambda)^2}\Big)+\frac{4a_{k0}(a_{k\bullet}-2a_{k0}+1)}{(a_{k0}-1+2\lambda)^2(a_{k\bullet}-1+4\lambda)}\Big)\Big|\\ &\leq\sum_{k=1}^{K}\Big(\frac{8a_{k1}|a_{k\bullet}-2a_{k1}+1|}{(a_{k1}-1)(a_{k\bullet}-1)^2}+\frac{4a_{k1}|a_{k\bullet}-2a_{k1}+1|}{(a_{k1}-1)^2(a_{k\bullet}-1)}\\ &+\frac{8a_{k0}|a_{k\bullet}-2a_{k0}+1|}{(a_{k0}-1)(a_{k\bullet}-1)^2}\Big)+\frac{4a_{k0}|a_{k\bullet}-2a_{k0}+1|}{(a_{k0}-1)^2(a_{k\bullet}-1)}\Big),\end{aligned}$$

where the last expression will be used as the smoothness constant L. We will not check if the function is completely convex, but we will find the area where the function is convex. On that area we can perform optimization, and find a suitable λ. We would like to have a better objective value than in the case of the unbiased estimator which value is reached at $\lambda = 0$. Therefore, we would like to have a convex area around zero point. We will show that $D''(0) > 0$. Using simple algebra, we obtain that:

$$D''(0) = \frac{4a_{k1}}{(a_{k1}-1)^2}+\frac{4a_{k0}}{(a_{k0}-1)^2}-\frac{16a_{k\bullet}}{(a_{k\bullet}-1)^2}.$$

We consider a function $f(x) = \frac{4x}{(x-1)^2} = \frac{4}{x-1}+\frac{4}{(x-1)^2}$. For $x > 1$, we have that the both summands from f are convex, so also f is convex. Using Jensen's inequality we have that:

$$\frac{4a_{k1}}{(a_{k1}-1)^2}+\frac{4a_{k0}}{(a_{k0}-1)^2}\geq 2\frac{4^{\frac{a_{k1}+a_{k0}}{2}}}{(\frac{a_{k1}+a_{k0}}{2}-1)^2}=\frac{16a_{k\bullet}}{(a_{k\bullet}-2)^2}>\frac{16a_{k\bullet}}{(a_{k\bullet}-1)^2},$$

for $a_{k\bullet} > 1$. Therefore, it follows that $D''(0) > 0$. From the facts that $D'(0) < 0$ and $D''(0) > 0$, we conclude that the gradient descent will have the right direction, so the iteration will converge to a positive value. Also, we do not have the whole interval where the function is convex, but since $D''(0) > 0$ and due to the continuity of the second derivative, we have that there exists a neighborhood of 0 where D is convex. A possible problem for this scenario is that we do not know if the function D will change the convexity before the local minimum appears. That possibility we cannot control, but however, we will try with the gradient descent method. Since we have calculated the first derivative and the bound of the second derivative, we have all prerequisites for the gradient descent.

Now, we need to calculate the same thing for the LOOCV mean squared error. We had that:

$$PE(\lambda)=\sum_{k=1}^{K}(a_{k1}(1-\hat{\pi}_{k1})^2+a_{k0}\hat{\pi}_{k0}^2)=\sum_{k=1}^{K}\frac{a_{k1}(a_{k0}+2\lambda)^2+a_{k0}(a_{k1}+2\lambda)^2}{(a_{k\bullet}+4\lambda-1)^2}.$$

We calculate and rewrite the derivative:

$$PE'(\lambda) = \sum_{k=1}^{K} \frac{8(\lambda((a_{k1}-a_{k0})^2 - a_{k\bullet}) - a_{k1}a_{k0})}{(a_{k\bullet}+4\lambda-1)^3}.$$
$$= \sum_{k=1}^{K} \frac{2((a_{k1}-a_{k0})^2 - a_{k\bullet})}{(a_{k\bullet}+4\lambda-1)^2} - \frac{2(a_{k\bullet}-1)((a_{k1}-a_{k0})^2 - a_{k\bullet})}{(a_{k\bullet}+4\lambda-1)^3} - \frac{8a_{k1}a_{k0}}{(a_{k\bullet}+4\lambda-1)^3}.$$

Also we have that:

$$PE'(0) = \sum_{k=1}^{K} -\frac{8a_{k1}a_{k0}}{(a_{k\bullet}-1)^3}.$$

So we have that $P'(0) < 0$, as in the deviance case. For the second derivative, we have that:

$$PE''(\lambda) = \sum_{k=1}^{K} -\frac{16((a_{k1}-a_{k0})^2 - a_{k\bullet})}{(a_{k\bullet}+4\lambda-1)^3} + \frac{24(a_{k\bullet}-1)((a_{k1}-a_{k0})^2 - a_{k\bullet})}{(a_{k\bullet}+4\lambda-1)^4} + \frac{96a_{k1}a_{k0}}{(a_{k\bullet}+4\lambda-1)^3},$$

and for the bound we have:

$$|PE'(\lambda)| \leq \sum_{k=1}^{K} \frac{16((a_{k1}-a_{k0})^2 - a_{k\bullet})}{(a_{k\bullet}-1)^3} + \frac{24(a_{k\bullet}-1)((a_{k1}-a_{k0})^2 - a_{k\bullet})}{(a_{k\bullet}-1)^4} + \frac{96a_{k1}a_{k0}}{(a_{k\bullet}-1)^3}.$$

We take the expression above for our smoothness constant L. Also for the convexity around zero point we have:

$$PE'(0) = \sum_{k=1}^{K} \frac{8(a_{k\bullet}-1)((a_{k1}-a_{k0})^2 - a_{k\bullet})}{(a_{k\bullet}+4\lambda-1)^4} + \frac{96a_{k1}a_{k0}}{(a_{k\bullet}+4\lambda-1)^3} > 0.$$

Therefore, we have convexity around zero. Also, as in the pervious case, we cannot guarantee that the minimum will be reached before the function changes its convexity. That is, we cannot guarantee that

$$\min_{\lambda>0}\{\lambda | f'(\lambda) = 0\} < \min_{\lambda>0}\{\lambda | f''(\lambda) = 0\}.$$

However, we hope that we will reach some suboptimal solution in the simulation study.

To sum up, we have constructed 4 ways of estimating the parameter λ, where two of them are explicit estimators, while the other two are an output of the optimization process. The criteria for their comparison will be the MSE risk.

5.2 Special cases

Before proceeding with general estimators, we will do a simulation study for the special cases. As we have already mentioned before, special cases are when $K = 1$ and when

sample sizes of each fold are equal. First, we consider the case when $K = 1$. We calculated an optimal λ for the MSE where we had that:

$$\lambda_{opt} = \frac{\pi_1(1-\pi_1)}{(1-2\pi_1)^2}.$$

Here, we will use the estimator obtained after replacing a probability with the unbiased estimator. The resulting estimator is of the following form:

$$\hat{\lambda}_1 = \frac{\hat{\pi}_1(1-\hat{\pi}_1)}{(1-2\hat{\pi}_1)^2}.$$

In this case, the optimum exists when the value of the unbiased estimator is not 0.5. If the unbiased estimator is equal to that value, we estimate λ as infinity, and we set the probability estimators to their limiting values. For the cross validation techniques, we can calculate the optimum explicitly. So, for the case of deviance we have that:

$$D'(\lambda) = 0 \implies \hat{\lambda}_2 = \frac{a_{10}(a_{10}-1) + a_{11}(a_{11}-1)}{2(a_{11}-a_{10})^2 - 2a_{1\bullet}}.$$

In case of the MSE, we have that

$$PE'(\lambda) = \frac{8(\lambda((a_{11}-a_{10})^2 - a_{1\bullet}) - a_{11}a_{10})}{(a_{1\bullet}+4\lambda-1)^3} = 0 \implies \hat{\lambda}_3 = \frac{a_{11}a_{10}}{(a_{11}-a_{10})^2 - a_{1\bullet}}.$$

Here, we need to distinguish two cases. If it holds that $(a_{11}-a_{10})^2 - a_{1\bullet} \leq 0$, then we have that $D'(\lambda) < 0$ and also $PE'(\lambda) < 0$. Further this means that both objectives are decreasing on the whole positive axis, which means that the optimal λ is at infinity. So, if we will have λ at infinity, we will set the probability estimators to their limiting values. Otherwise, there exist a unique optimum on the positive axis.

The simulations are done using programming language PYTHON with version 3.6 [18]. First, for a given sample size $a_{1\bullet}$ and a probability π_1, we generate a sample from the Bernoulli distribution using scientific package NUMPY [9]. In all further simulations, the same language and package will be used. After we created a sample, we compute the value of an estimator, and we calculate the squared difference between that value and the real probability. We will call it the basic MSE. We repeat this process 100000 times, and we obtain 100000 basic MSEs. Then we average all of them and we obtain the empirical MSE. Due to the weak law of large numbers and since we did a large number of repetitions, this empirical MSE is a good approximation of the real MSE. During simulations in this case, from the estimators which are dependent on λ, it is easy to check that for $K = 1$ it holds: $\hat{\pi}_1 = \hat{\pi}_1^{2,\lambda} = \hat{\pi}_1^{3,\lambda}$. So, we calculate only $\hat{\pi}_1$ and $\hat{\pi}_1^{\lambda}$.

Now, we provide the results of the simulations. On each figure, we have the squared root of the mean squared error (RMSE). We have this for the unbiased estimator and

estimators $\hat{\pi}_1^{\lambda}$ with three different λ estimates. We have also a value of optimal λ with real probabilities. We introduce it for the comparison reasons. That value is denoted with "lambda oracle" at the legend. For fixed $a_{1\bullet}$, on the y-axis we have RMSE while on the $x-$axis we have probability. In the captions of the images, we have value of $a_{1\bullet}$ used to create the artificial sample.

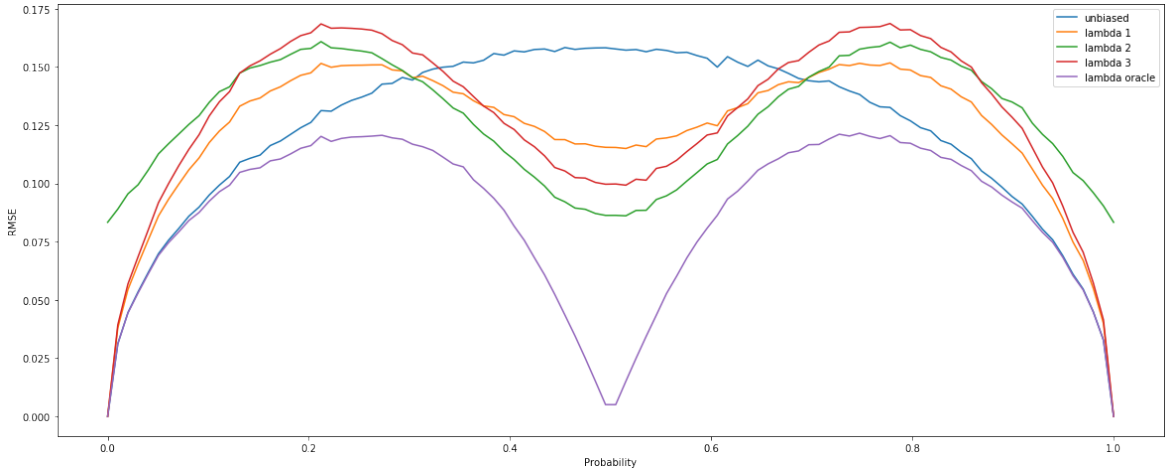


Figure 3: $a_{1\bullet} = 10$

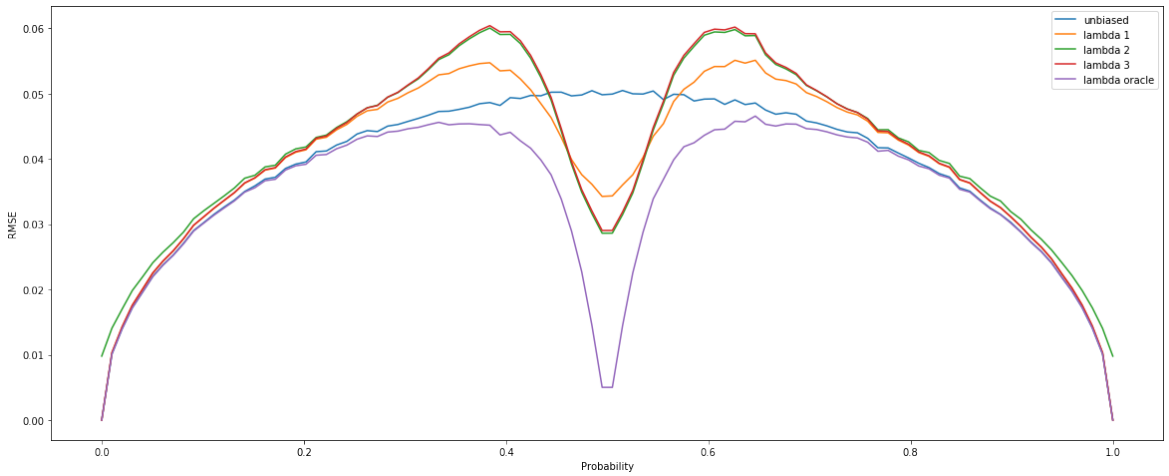


Figure 4: $a_{1\bullet} = 100$

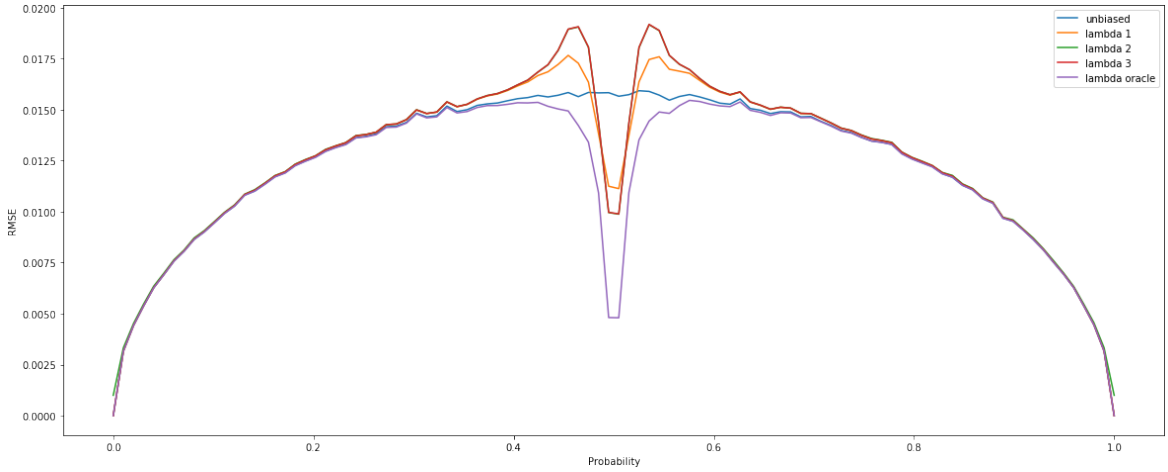


Figure 5: $a_{1\bullet} = 1000$

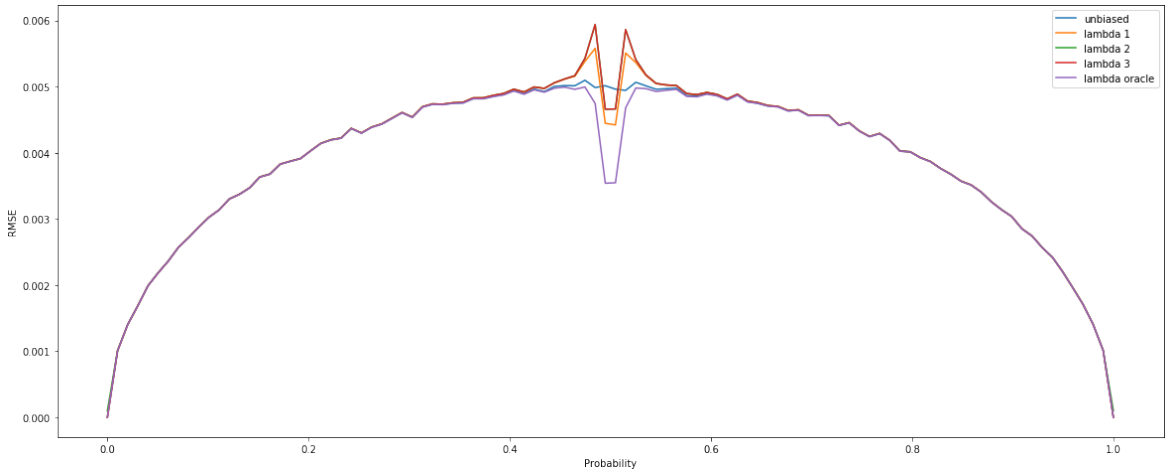


Figure 6: $a_{1\bullet} = 10000$

From the results we see that we did not succeed to improve the unbiased estimator in general. On every figure, there is a probability interval where $\hat{\pi}_1^{\lambda}$ is better than the unbiased estimator. That interval is around 0.5. If we are far from 0.5, we have that unbiased estimator is better for smaller fold size. This is hard to interpret since it is hard to see what is happening with bias or with variance of the estimator after we add λ. The possible interpretations are:

- If probabilities are far from 0.5, the unbiased estimator has smaller variance, since the variance is proportional to $\pi_1(1 - \pi_1)$. So λ may introduce additional variability. Of course, if we add λ we do not have an unbiased estimator anymore. So, possibly the increase of the bias caused a worse performance of π_1^{λ}.

- If the probabilities are close to 0.5, we have obvious increase of the variability of the unbiased estimator. Again, the unbiased estimator has the highest variance

for $\pi_1 = 0.5$. So adding λ we reduced this high variance.

Second thing we may notice is that if we increase the fold size, the estimators tend to be the same. How this can be interpreted? In the estimator $\hat{\pi}_1^{\lambda}$, λ appears in both numerator and denominator. Estimators of λ are ratios of polynomials of the same degree. So, we may interpret that the increase of the fold size, does not have big impact on λ; λ stays nearly the same. On the other side, other values in the numerator and in the denominator will increase (a_{11} and $a_{1\bullet}$). So, if we increase the fold sizes, λ's impact on the estimator decreases. So π_1^{λ} tends to be the same as the unbiased one.

As another special case we have identified the one where all folds are of the same size, which means it holds that $a_{1\bullet} = \cdots = a_{K\bullet} = a_{\bullet}$. For this specific example, we can construct all 3 estimators mentioned above dependent on λ, but we can obtain only 2 different λ estimates. As we wrote above, the optimal value of the MSE can be achieved for:

$$\lambda_{opt} = \frac{\sum_{k=1}^{K} \pi_k(1-\pi_k)}{\sum_{k=1}^{K}(1-2\pi_k)^2}.$$

Using the same idea as for the previous special case, the obtained estimator is:

$$\hat{\lambda}_1 = \frac{\sum_{k=1}^{K} \hat{\pi}_k(1-\hat{\pi}_k)}{\sum_{k=1}^{K}(1-2\hat{\pi}_k)^2}.$$

We cannot apply the same trick to the deviance objective function since a_{k1} values are appearing in the denominator of the expression, so we cannot find the exact expression in this case. For the MSE objective in cross validation case, we have the optimal solution as:

$$\hat{\lambda}_2 = \frac{\sum_{k=1}^{K} a_{k1}a_{10}}{\sum_{k=1}^{K}(a_{k1}-a_{k0})^2 - Ka_{\bullet}}.$$

First we examine the case when all probabilities are equal. That is the most trivial case and we will provide plots like we did for the last special case. In this case, we will not change the fold size, since we have noticed the same behavior as before: the estimators tends to be the same for too large fold sizes. To apply estimators to the data from the Brown's article, the most representative case is to have $a_{\bullet} = 500$, since the number of hits is around 500 in average. We provide 9 graphs, for 3 different values of K. We take $K \in \{5, 50, 500\}$. For each value, we have three graphs for three different estimators. On the y-axis we have RMSE, while on the x-axis we have probabilities. The values of λ_{opt} are denoted as "lambda oracle" at the legend.

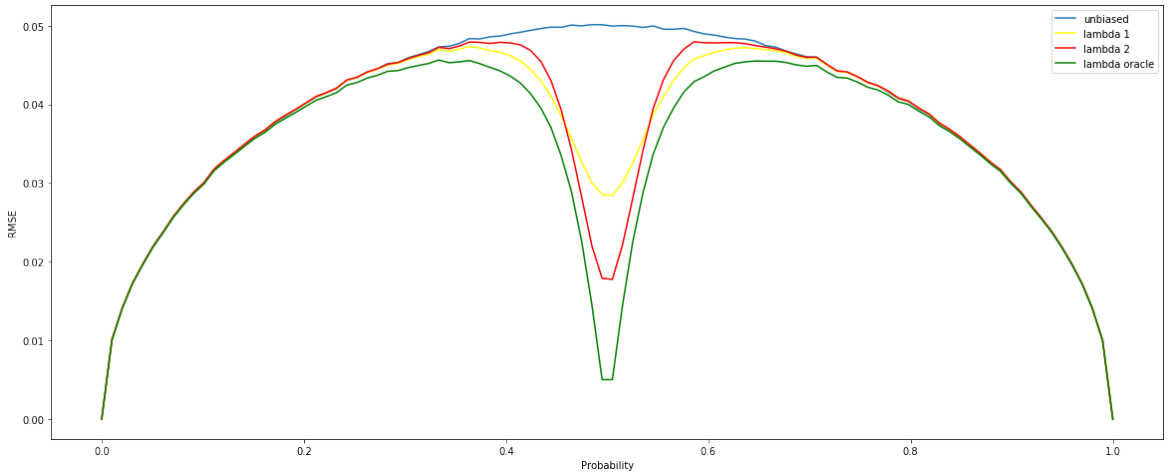


Figure 7: Results for $\hat{\pi}_k^{\lambda}$ for $K = 5, a_{k\bullet} = 100$

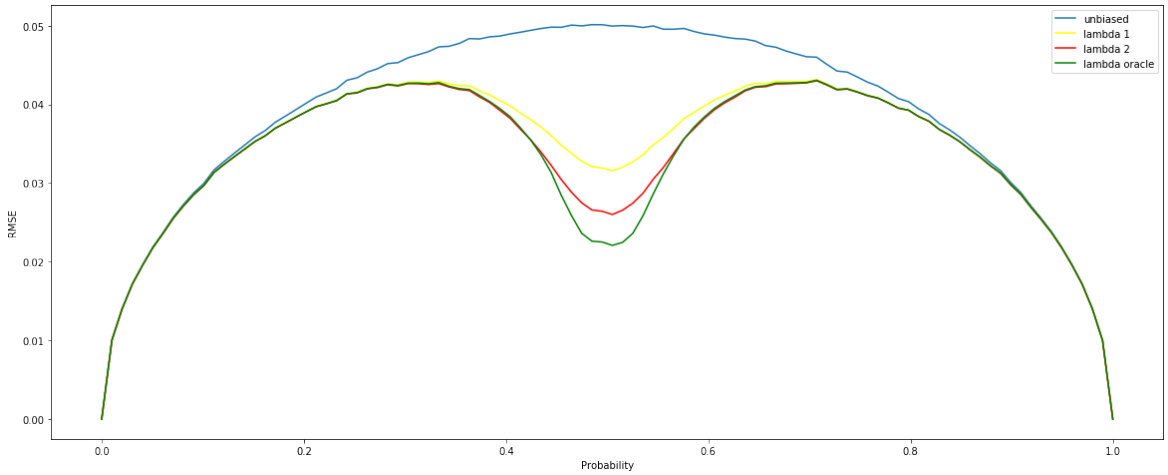


Figure 8: Results for $\hat{\pi}_k^{2,\lambda}$ for $K = 5, a_{1\bullet} = 100$

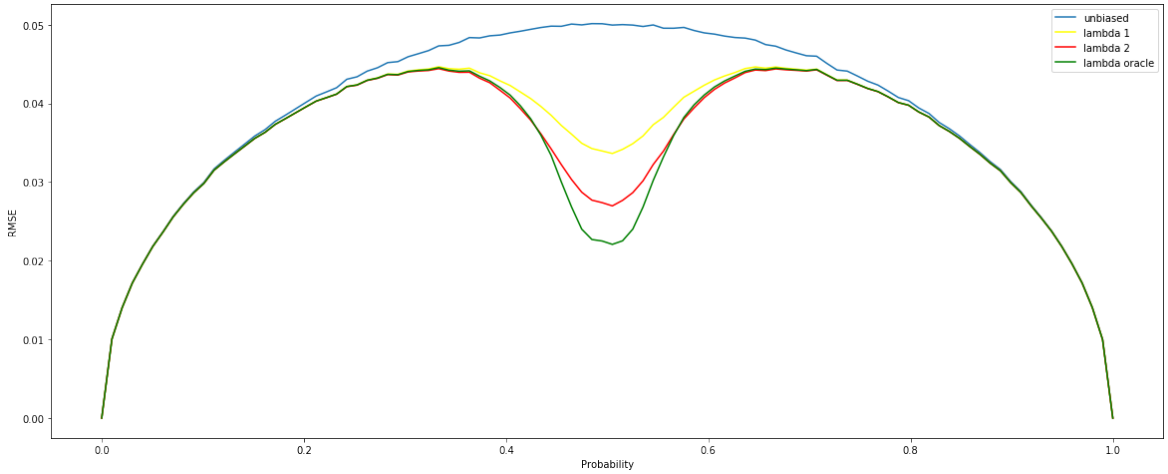


Figure 9: Results for $\hat{\pi}_k^{3,\lambda}$ for $K = 5, a_{1\bullet} = 100$

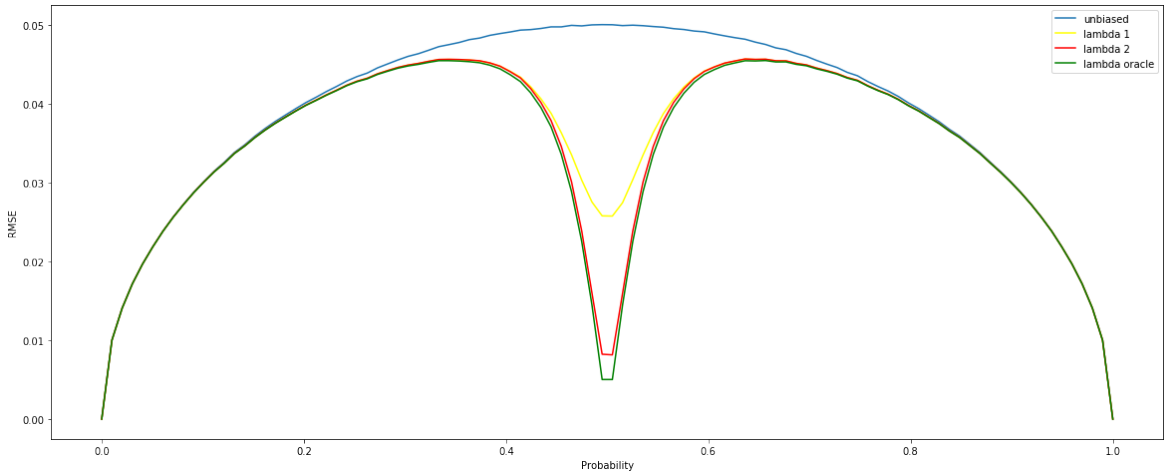


Figure 10: Results for $\hat{\pi}_k^{\lambda}$ for $K = 50, a_{k\bullet} = 100$

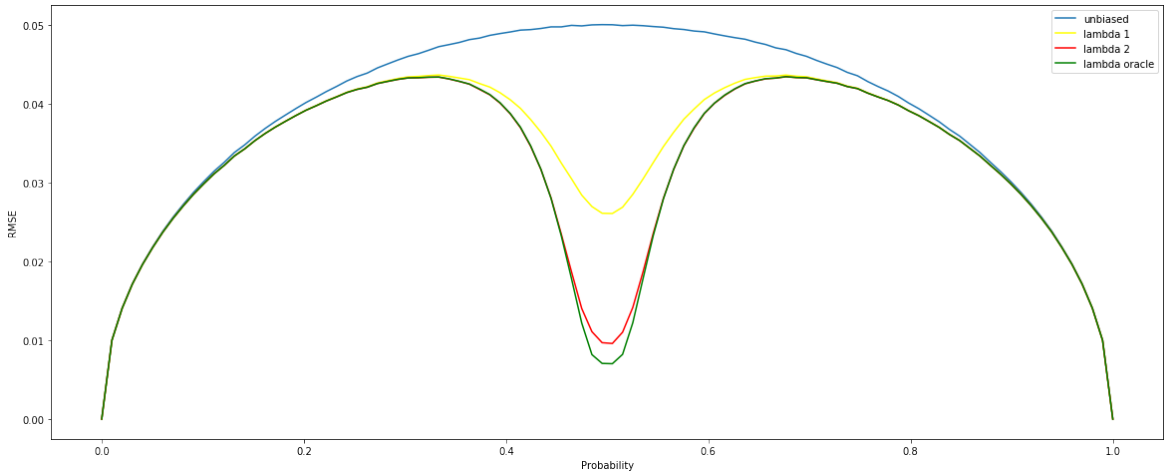


Figure 11: Results for $\hat{\pi}_k^{2,\lambda}$ for $K = 50, a_{1\bullet} = 100$

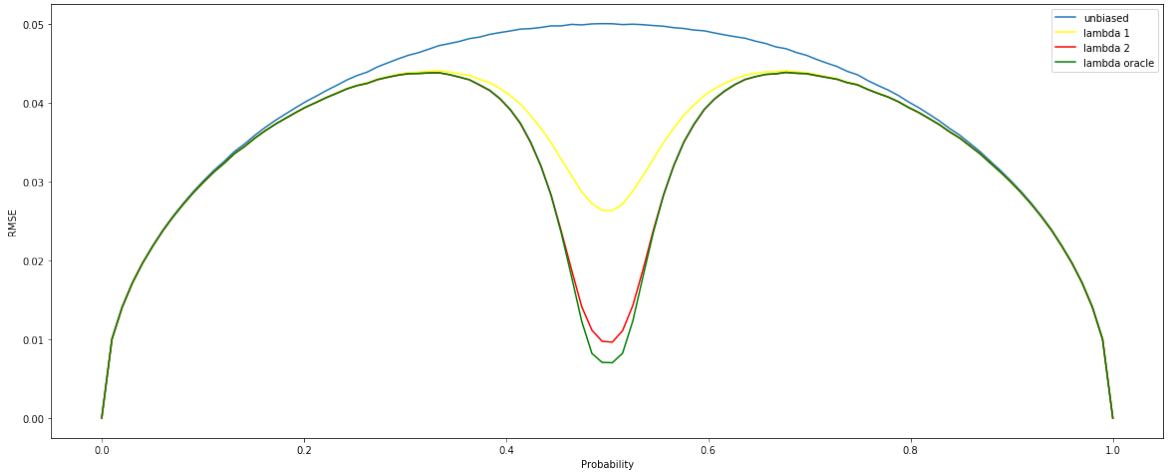


Figure 12: Results for $\hat{\pi}_k^{3,\lambda}$ for $K = 50, a_{1\bullet} = 100$

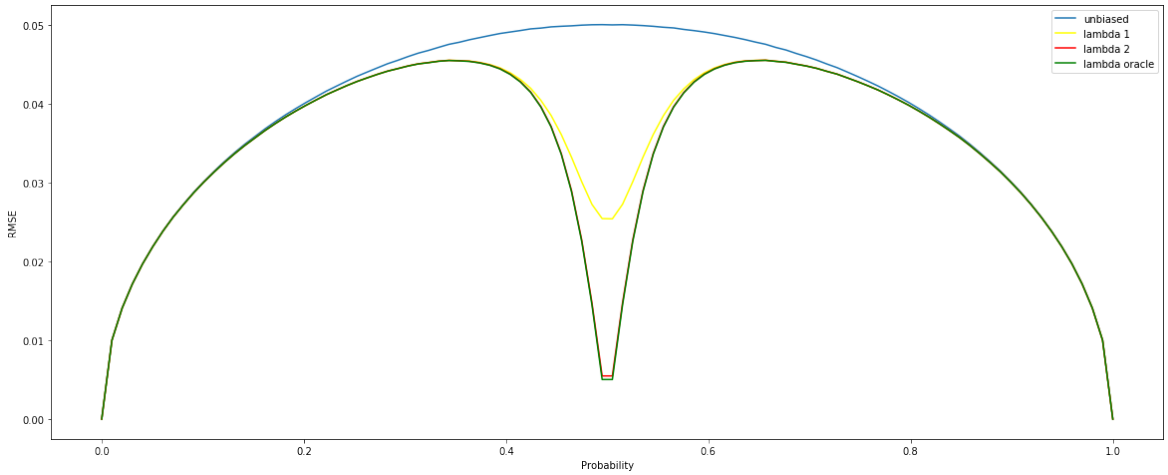


Figure 13: Results for $\hat{\pi}_k^{\lambda}$ for $K = 500, a_{1\bullet} = 100$

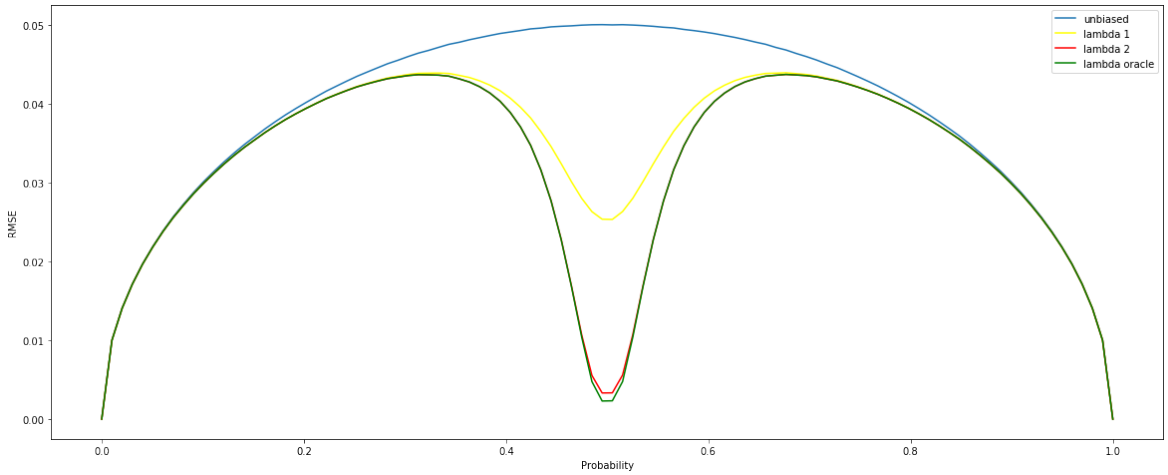


Figure 14: Results for $\hat{\pi}_k^{2,\lambda}$ for $K = 500, a_{1\bullet} = 100$

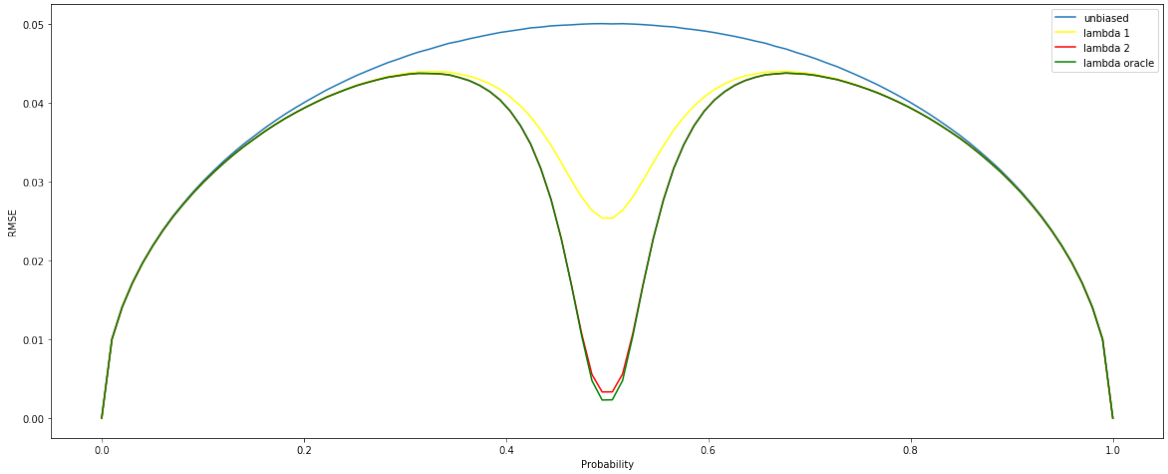


Figure 15: Results for $\hat{\pi}_k^{3,\lambda}$ for $K = 500, a_{1\bullet} = 100$

Finally, we may say that we have a success. Namely, both of $\pi_k^{2,\lambda}$ and $\pi_k^{3,\lambda}$ performed better for each K and for both λ. That is especially visible when the probability is close to 0.5. In that case, the differences are huge; somewhere even 5 times. Also, at every image we can see better performance of $\hat{\lambda}_2$ than $\hat{\lambda}_1$, which is unexpected since $\hat{\lambda}_1$ is the estimator for optimal λ. Of course, we formed it straightforward, so it may be an estimator with huge MSE for itself. For π_k^{λ}, we have the feeling that it also overperforms the unbiased one. For $K = 50$, hat is not true for extreme probabilities. It can be shown that on the edges of the graph π_k^{λ} is smaller than the unbiased one. However, when we increase K, it becomes better, and for huge K, π_k^{λ} became better than the unbiased one. If compare all estimators between themselves, we may see that estimator $\pi_k^{2,\lambda}$ is giving the best results, then $\pi_k^{3,\lambda}$, and the worst is π_k^{λ}. This is also unexpected, since λ_1 is optimal solution for MSE of π_k^{λ}, so we would expect better performance for π_k^{λ} than for other estimators, for at least $\hat{\lambda}_1$.

Now we will try to do some analysis when the probabilities are not the same. Every probability is a parameter for itself. Therefore, for so many parameters it is hard to provide a general view of every possible situation. We remain to build our intuition based on the special cases. Here we provide simulation studies for $K = 3$, and we will try with six probability vectors: $(0.05, 0.05, 0.05)$, $(0.05, 0.05, 0.45)$, $(0.05, 0.45, 0.45)$, $(0.05, 0.2, 0.2)$, $(0.2, 0.2, 0.45)$, $(0.05, 0.2, 0.45)$, together with three different sizes of $a_\bullet$. In each cell, we have $100 \cdot RMSE$ for a particular probability estimator combined with a particular λ estimator. We use $100 \cdot RMSE$ for better visibility of results.

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	6.908	6.908	6.908
$\hat{\pi}^{\lambda}$	7.283	7.378	6.831
$\hat{\pi}^{2,\lambda}$	6.609	6.564	6.707
$\hat{\pi}^{3,\lambda}$	6.743	6.716	6.802

$\boldsymbol{\pi} = (0.05, 0.05, 0.05), a_\bullet = 10$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	5.121	5.121	5.121
$\hat{\pi}^{\lambda}$	5.377	5.442	5.068
$\hat{\pi}^{2,\lambda}$	4.917	4.888	4.983
$\hat{\pi}^{3,\lambda}$	5.009	4.990	5.049

$\boldsymbol{\pi} = (0.05, 0.05, 0.05), a_\bullet = 100$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	4.200	4.200	4.200
$\hat{\pi}^{\lambda}$	4.409	4.461	4.157
$\hat{\pi}^{2,\lambda}$	4.035	4.011	4.088
$\hat{\pi}^{3,\lambda}$	4.109	4.094	4.141

$\boldsymbol{\pi} = (0.05, 0.05, 0.05), a_\bullet = 1000$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	6.506	6.506	6.506
$\hat{\pi}^{\lambda}$	6.640	6.739	6.311
$\hat{\pi}^{2,\lambda}$	6.325	6.363	6.241
$\hat{\pi}^{3,\lambda}$	6.331	6.315	6.289

$\boldsymbol{\pi} = (0.05, 0.05, 0.45), a_\bullet = 10$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	6.014	6.014	6.014
$\hat{\pi}^{\lambda}$	6.130	6.216	5.844
$\hat{\pi}^{2,\lambda}$	5.857	5.889	5.783
$\hat{\pi}^{3,\lambda}$	5.861	5.848	5.824

$\boldsymbol{\pi} = (0.05, 0.05, 0.45), a_\bullet = 100$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	5.507	5.507	5.507
$\hat{\pi}^{\lambda}$	5.613	5.691	5.353
$\hat{\pi}^{2,\lambda}$	5.364	5.393	5.297
$\hat{\pi}^{3,\lambda}$	5.368	5.355	5.335

$\boldsymbol{\pi} = (0.05, 0.05, 0.45), a_\bullet = 1000$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	7.214	7.214	7.214
$\hat{\pi}^{\lambda}$	7.104	7.293	6.716
$\hat{\pi}^{2,\lambda}$	6.835	6.932	6.668
$\hat{\pi}^{3,\lambda}$	6.937	6.976	6.791

$\boldsymbol{\pi} = (0.05, 0.45, 0.45), a_\bullet = 10$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	6.912	6.912	6.912
$\hat{\pi}^{\lambda}$	6.810	6.983	6.454
$\hat{\pi}^{2,\lambda}$	6.562	6.651	6.409
$\hat{\pi}^{3,\lambda}$	6.657	6.692	6.523

$\boldsymbol{\pi} = (0.05, 0.45, 0.45), a_\bullet = 100$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	6.532	6.532	6.532
$\hat{\pi}^{\lambda}$	6.436	6.599	6.101
$\hat{\pi}^{2,\lambda}$	6.203	6.286	6.058
$\hat{\pi}^{3,\lambda}$	6.292	6.325	6.166

$\boldsymbol{\pi} = (0.05, 0.45, 0.45), a_\bullet = 1000$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	7.104	7.104	7.104
$\hat{\pi}^{\lambda}$	7.071	7.259	6.671
$\hat{\pi}^{2,\lambda}$	6.664	6.708	6.541
$\hat{\pi}^{3,\lambda}$	6.806	6.813	6.707

$\boldsymbol{\pi} = (0.05, 0.2, 0.2), a_\bullet = 10$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	6.854	6.854	6.854
$\hat{\pi}^{\lambda}$	6.825	7.002	6.446
$\hat{\pi}^{2,\lambda}$	6.438	6.479	6.323
$\hat{\pi}^{3,\lambda}$	6.573	6.579	6.479

$\boldsymbol{\pi} = (0.05, 0.2, 0.2), a_\bullet = 100$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	6.570	6.570	6.570
$\hat{\pi}^{\lambda}$	6.542	6.712	6.180
$\hat{\pi}^{2,\lambda}$	6.173	6.212	6.062
$\hat{\pi}^{3,\lambda}$	6.301	6.307	6.212

$\boldsymbol{\pi} = (0.05, 0.2, 0.2), a_\bullet = 1000$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	7.384	7.384	7.384
$\hat{\pi}^{\lambda}$	7.302	7.595	6.817
$\hat{\pi}^{2,\lambda}$	6.883	6.924	6.732
$\hat{\pi}^{3,\lambda}$	7.000	6.992	6.847

$\boldsymbol{\pi} = (0.2, 0.2, 0.45), a_\bullet = 10$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	7.210	7.210	7.210
$\hat{\pi}^{\lambda}$	7.133	7.412	6.668
$\hat{\pi}^{2,\lambda}$	6.733	6.772	6.588
$\hat{\pi}^{3,\lambda}$	6.842	6.834	6.696

$\boldsymbol{\pi} = (0.2, 0.2, 0.45), a_\bullet = 100$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	6.974	6.974	6.974
$\hat{\pi}^{\lambda}$	6.900	7.169	6.452
$\hat{\pi}^{2,\lambda}$	6.515	6.552	6.374
$\hat{\pi}^{3,\lambda}$	6.619	6.612	6.479

$\boldsymbol{\pi} = (0.2, 0.2, 0.45), a_\bullet = 1000$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	7.427	7.427	7.427
$\hat{\pi}^{\lambda}$	7.347	7.625	6.874
$\hat{\pi}^{2,\lambda}$	6.962	7.011	6.809
$\hat{\pi}^{3,\lambda}$	7.050	7.040	6.900

$\boldsymbol{\pi} = (0.05, 0.2, 0.45), a_\bullet = 10$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	7.267	7.267	7.267
$\hat{\pi}^{\lambda}$	7.190	7.457	6.734
$\hat{\pi}^{2,\lambda}$	6.820	6.867	6.672
$\hat{\pi}^{3,\lambda}$	6.904	6.894	6.759

$\boldsymbol{\pi} = (0.05, 0.2, 0.45), a_\bullet = 100$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	λ_{opt}
$\hat{\pi}$	7.068	7.068	7.068
$\hat{\pi}^{\lambda}$	6.993	7.253	6.551
$\hat{\pi}^{2,\lambda}$	6.634	6.680	6.490
$\hat{\pi}^{3,\lambda}$	6.715	6.706	6.575

$\boldsymbol{\pi} = (0.05, 0.2, 0.45), a_\bullet = 1000$

Table 1: Results for equal folds when $K = 3$

We can see from the provided results, in almost all cases, we have that the estimator $\hat{\pi}^{2,\lambda}$ showed the best results, combined with $\hat{\lambda}_1$. This is surprising given that $\hat{\lambda}_1$ was

made based on the optimal λ value for the estimator $\hat{\pi}^{\lambda}$. On the other side, $\hat{\pi}^{\lambda}$ did not improve the unbiased one in the cases when we had vectors with 2 or 3 small probabilities, but it is better in the cases with larger probabilities. For the estimator $\hat{\pi}^{3,\lambda}$ we can see that it is usually better than $\hat{\pi}^{\lambda}$, while it is worse than $\hat{\pi}^{2,\lambda}$. Also the characteristic for this one is that, for larger probabilities, it works better with $\hat{\lambda}_2$ estimator than with $\hat{\lambda}_1$.

5.3 General case

In this section we provide a simulation study for general case. So, we will not assume anymore that some probabilities or some folds are equal. In this case, we face with a lot of parameters to be tested. We may have a situation where $K = 100$. So we have to test for 100 different probabilities and 100 fold sizes. That is in total 200 parameters. And each of these parameters should be tested for different values. So we would need 200-dimensional euclidean space to present the results. That is, of course, impossible, so we need to find smarter strategy for testing. Let us now return to our practical motivation. We will describe the data from the Brown's article. We have a data from 929 players. Many of those players had just a few hits. The performance from a small number of hits does not illustrate the real ability of the player. So Brown decided to remove those players with a small number of hits. The threshold was 11, so we include only players who have 11 or more hits in both half-seasons for testing. After filtering, we remain with 499 players. So for testing, we would like to simulate artificial data to be similar to the data from the article. For the total number of hits in the first half-season, we provide a plot of the distribution on Figure 16. On Figure 16, we have

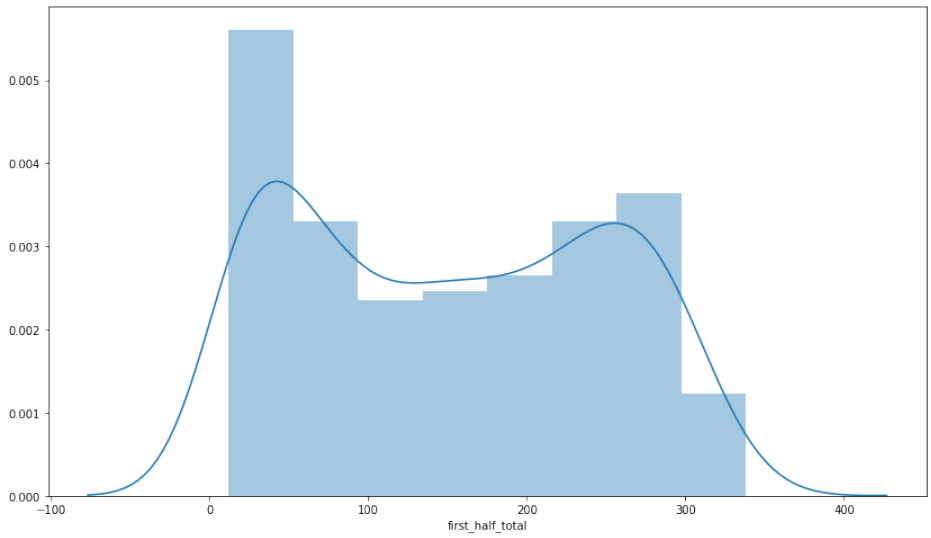


Figure 16: Distribution of the hits in the first half-season

a distribution from which we would like to simulate our data. We cannot recognize any known distribution from the image. We notice that we have a bimodal distribution, where each modal has a similar shape to the normal distribution, more or less. So to model the distribution, we will use a Gaussian Mixture model. Gaussian mixture is a probability distribution which probability distribution function (PDF) is weighted sum of PDF's of normals. Here, we will assume that we have a sum of two normals. To estimate parameters of such distribution we use the EM algorithm [6]. After we found the parameters, we use them to simulate the values for the folds. To simulate

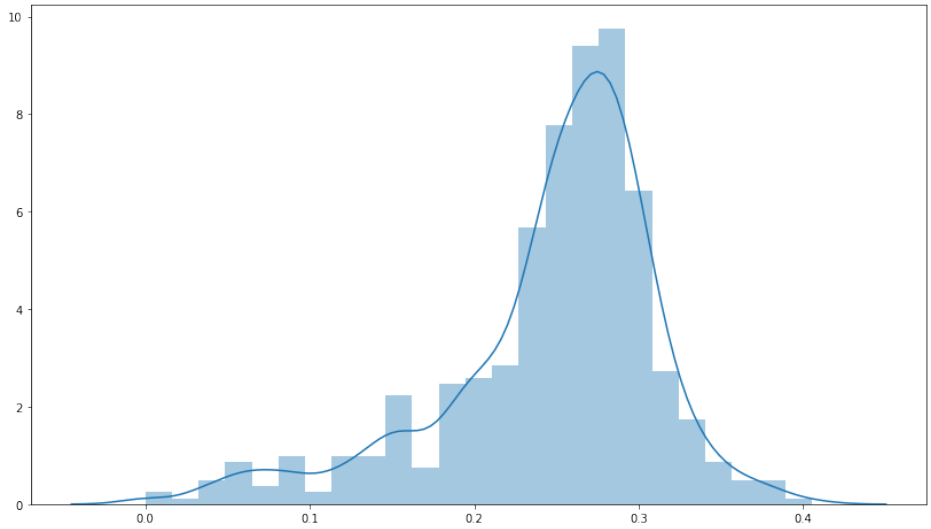


Figure 17: Distribution of the ratios of successful hits in the first half-season

probabilities, we consider ratios of the number of successful hits and the total number of hits. We obtain a vector of unbiased estimators of probabilities of successful hits. The distribution of such probabilities is shown at Figure 17. Again, we cannot identify the distribution from the image. We tried with a Beta distribution, but it did not fit the best. We have tried Gaussian Mixture model here too. Surprisingly, it gave very good results. It even fits better than in the previous case for folds. So again, we fitted Gaussian Mixture model for two normal distributions, and we used estimated parameters to generate the probabilities.

Now we provide 10 tabels for 10 different samples, sampled from Gausian mixture models for both, folds and probabilities. In every cell, for a particular probability estimator, and a particular λ estimator, we have $100 \cdot RMSE$ for better view of results. We have used estimators of λ described in Section 5.1.

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$	$\hat{\lambda}_4$	λ^*
$\hat{\pi}$	4.815	4.815	4.815	4.815	4.815
$\hat{\pi}^{\lambda}$	4.797	4.794	4.808	4.676	4.797
$\hat{\pi}^{2,\lambda}$	4.776	4.769	4.543	4.335	4.776
$\hat{\pi}^{3,\lambda}$	4.779	4.773	4.543	4.336	4.779

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$	$\hat{\lambda}_4$	λ^*
$\hat{\pi}$	4.962	4.962	4.962	4.962	4.962
$\hat{\pi}^{\lambda}$	4.940	4.936	4.945	4.752	4.937
$\hat{\pi}^{2,\lambda}$	4.917	4.908	4.747	4.398	4.913
$\hat{\pi}^{3,\lambda}$	4.920	4.912	4.747	4.399	4.917

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$	$\hat{\lambda}_4$	λ^*
$\hat{\pi}$	4.955	4.955	4.955	4.955	4.955
$\hat{\pi}^{\lambda}$	4.934	4.931	4.928	4.766	4.931
$\hat{\pi}^{2,\lambda}$	4.911	4.904	4.656	4.404	4.908
$\hat{\pi}^{3,\lambda}$	4.915	4.908	4.656	4.406	4.912

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$	$\hat{\lambda}_4$	λ^*
$\hat{\pi}$	5.004	5.004	5.004	5.004	5.004
$\hat{\pi}^{\lambda}$	4.984	4.980	4.990	4.822	4.981
$\hat{\pi}^{2,\lambda}$	4.957	4.950	4.705	4.418	4.953
$\hat{\pi}^{3,\lambda}$	4.963	4.956	4.706	4.421	4.959

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$	$\hat{\lambda}_4$	λ^*
$\hat{\pi}$	5.012	5.012	5.012	5.012	5.012
$\hat{\pi}^{\lambda}$	4.993	4.989	5.000	4.849	4.990
$\hat{\pi}^{2,\lambda}$	4.965	4.958	4.666	4.428	4.962
$\hat{\pi}^{3,\lambda}$	4.971	4.964	4.667	4.431	4.968

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$	$\hat{\lambda}_4$	λ^*
$\hat{\pi}$	5.050	5.050	5.050	5.050	5.050
$\hat{\pi}^{\lambda}$	5.030	5.026	5.039	4.874	5.027
$\hat{\pi}^{2,\lambda}$	5.001	4.993	4.724	4.440	4.998
$\hat{\pi}^{3,\lambda}$	5.007	4.999	4.725	4.443	5.004

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$	$\hat{\lambda}_4$	λ^*
$\hat{\pi}$	5.038	5.038	5.038	5.038	5.038
$\hat{\pi}^{\lambda}$	5.017	5.014	5.028	4.860	5.015
$\hat{\pi}^{2,\lambda}$	4.987	4.979	4.733	4.426	4.985
$\hat{\pi}^{3,\lambda}$	4.993	4.986	4.734	4.429	4.991

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$	$\hat{\lambda}_4$	λ^*
$\hat{\pi}$	5.016	5.016	5.016	5.016	5.016
$\hat{\pi}^{\lambda}$	4.996	4.993	5.012	4.851	4.994
$\hat{\pi}^{2,\lambda}$	4.967	4.959	4.710	4.416	4.965
$\hat{\pi}^{3,\lambda}$	4.973	4.965	4.712	4.418	4.971

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$	$\hat{\lambda}_4$	λ^*
$\hat{\pi}$	5.010	5.010	5.010	5.010	5.010
$\hat{\pi}^{\lambda}$	4.990	4.987	5.004	4.851	4.988
$\hat{\pi}^{2,\lambda}$	4.961	4.954	4.691	4.420	4.959
$\hat{\pi}^{3,\lambda}$	4.967	4.960	4.693	4.423	4.965

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$	$\hat{\lambda}_4$	λ^*
$\hat{\pi}$	5.015	5.015	5.015	5.015	5.015
$\hat{\pi}^{\lambda}$	4.996	4.992	5.011	4.858	4.994
$\hat{\pi}^{2,\lambda}$	4.967	4.959	4.690	4.422	4.965
$\hat{\pi}^{3,\lambda}$	4.972	4.965	4.691	4.425	4.971

Table 2: Results for general case

In most cases, we have improved the unbiased estimator. The best results are obtained when we are using $\hat{\pi}^{2,\lambda}$ and $\hat{\pi}^{3,\lambda}$ combined with LOOCV constructed by using mean squared error as a loss function. In that case, in every table we have improvement in

every cell for more than 10%. For different probability estimators, $\hat{\pi}^{2,\lambda}$ and $\hat{\pi}^{3,\lambda}$ have significantly better peformance than $\hat{\pi}^{\lambda}$. They always overperform the unbiased estimator, while $\hat{\pi}^{\lambda}$ under-performs in some cases. This may be because in the numerator of $\hat{\pi}^{2,\lambda}$ and $\hat{\pi}^{3,\lambda}$ we have an additional multiplier of λ which is adding more regularization to the estimator. Also we can see that $\hat{\pi}^{2,\lambda}$ combined with $\hat{\lambda}_4$ gave better result than the best possible $\hat{\pi}^{\lambda}$ (the one with λ_{opt}). This is showing us that the encoding of the categorical data has large impact on the estimation, so maybe for some other encoding we would be able to get an even better estimator. Comparing different λ, we may see that those obtained using cross validation are giving better results. We noticed that when we increase K, from the reason mentioned above, $\hat{\lambda}_1$ and $\hat{\lambda}_2$ remain the same, independent of K. That is because they are ratios of the polynomials of the same degree, so that ratio is independent of K. For the cross validation estimators of λ, we noticed that they are increasing while K is increasing. So for a large K, like we had $K = 500$, we have that $\hat{\lambda}_3$ and $\hat{\lambda}_4$ are significantly larger than $\hat{\lambda}_1$ and $\hat{\lambda}_2$, so they have more impact on the estimates.

So, with the satisfiable results from the previous sections, where we have improved the unbiased estimator, we proceed to apply the new estimator on the real data.

5.4 Application on batting averages

In this section we will apply the obtained probability estimators and λ estimators on the new data, hoping that we will improve the Brown's result. As we have already mentioned we have $K = 499$ different players for prediction. For the comparison between different results, Brown did not use (1.4) directly, but he used the following error:

$$TSE^* = \frac{TSE(\hat{R})}{TSE(\hat{R}_0)},$$

where $\hat{R}_0$ stands for the unbiased estimator. So if we have an improvement, we expect to have the values of the above risk less than 1. As more as it is less, we have better result. As additional information in this case, we use the number of total hits for each player in second half-season. If we assume that those hits are known quantity for us, we can use them to try to improve our result. For that we consider the result from Section 3.5. There, we have constructed the loss function which can be used in the case when we know the number of performed prediction for each fold. This is exactly the case here, where we know the number of hits in advance. In that section, we have constructed a particular λ which is suboptimal for that particular case. We will use it here also as additional help. First, we need to find an estimator of it. Again we have that $\pi_k(1 - \pi_k)$ is appearing in the expression. So we have the two options to estimate

it as we described in the previous sections. From the results from the previous section, we saw that the unbiased estimator of the variance gave better results. So, we will use only that approach here. For the estimator of the suboptimal λ from Section 3.5 we have

$$\hat{\lambda}_5 = \frac{\sum_{k=1}^{K} a_{k\bullet\bullet}\left(\frac{a_{k1}}{a_{k\bullet}^2(a_{k\bullet}-1)} - \frac{a_{k1}^2}{a_{k\bullet}^3(a_{k\bullet}-1)}\right)}{a_{\bullet\bullet}\left(2\sqrt{K\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^4}} + \sum_{k=1}^{K}\frac{1}{a_{k\bullet}^2} + 2\sqrt{\sum_{k=1}^{K}\frac{1}{a_{k\bullet}}}\sqrt{\sum_{k=1}^{K}\frac{1}{a_{k\bullet}^5}} + \sum_{k=1}^{K}\frac{1}{a_{k\bullet}^3}\right)},$$

We add new $\hat{\lambda}_5$ to the four other estimators form the previous section. We apply three probability estimators together with five λ estimators on the real dataset. The results are given in Table 3.

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$	$\hat{\lambda}_4$	$\hat{\lambda}_5$
$\hat{\pi}$	1.000	1.000	1.000	1.000	1.000
$\hat{\pi}^{\lambda}$	0.999	0.999	1.042	1.011	1.000
$\hat{\pi}^{2,\lambda}$	0.998	0.998	0.959	0.973	1.000
$\hat{\pi}^{3,\lambda}$	0.998	0.998	0.959	0.973	1.000

Table 3: Results obtained for the data from the Brown's article

.

As we can see that again the cross validation methods gave the best results. Surprisingly, $\hat{\lambda}_5$ did not provide any improvement, despite the fact that it has an additional information. Again, this can be interpreted that $\hat{\lambda}_5$ is too small for a large K to have any significant impact on the estimators. Additional reason is that it is not the optimal, but a suboptimal solution. So, it may remain the same even if K is increasing. Now, the final moment is the comparison with the result from the Brown's article. Using normal approximation described in the first section, together with the James-Stein estimator, Brown achieved $TSE^* = 0.54$. This result is incomparably better than any result that we have achieved. So, we did not succeed to improve the Brown's result.

6 Conclusion and future work

In this thesis, we were solving the problem of estimating the parameter for multiple Bernoulli distributions at once. We wanted to improve the mean squared error of the unbiased estimator. We introduced certain ways to obtain different estimators for the vector of parameters using logistic regression with ridge penalization. To use regression on categorical data, we had to encode categories in a proper way. So, we have presented three ways how to do that. Then we have constructed a ridge regression model for each of those encodings, which gave us three different regression models. From those models, we have expressed three estimators. As we have seen from the final results, some estimators gave significantly better results than the others. That implies that the way of encoding has a huge impact on the estimation. We have presented different encoding as a linear mapping of the standard $1-0$ encoding. So, for the future work in this case, we propose to try to improve the result using different linear maps, even more complex than we had. That will have a huge impact on the penalization expression, which will give different estimators.

After we have constructed our estimators, we had that all of them are dependent on the penalization coefficient λ. So, our task was to find a proper λ to improve our estimators. The first method we have used is expressing MSE as a function of λ. After we did that, we obtained $MSE(\lambda)$ as a function of one variable. We wanted to obtain the optimal λ which minimizes that function. $MSE(\lambda)$ was dependent on the real probability values, so such function is unknown in practice. So, we cannot optimize it directly. Instead of that, we found a value, which is not optimal, but still better than the case without penalization ($\lambda = 0$). We have tested that value. In majority cases, it gave better results, but in few cases, it did not. We noticed that when we increase the number of folds K, such λ performs worse. The conclusion was that the impact of it stagnates for large K. One possible solution for this case is that, we may estimate $MSE(\lambda)$, and using numerical methods, we can find the optimal λ from estimated MSE function. That solution would probably have a larger impact on the probability estimator for large K.

Another approach we have used for estimating λ is the cross validation. We have explained that approach, emphasizing its usage for determining λ. We used LOOCV to obtain the objective function, which minimization will provide us a suitable λ as the

estimator. That objective function was constructed using two different loss functions: deviance and mean squared error. This approach gave us the best result among all for both loss functions. With cross validation we have achieved 15% of improvement in MSE in some cases. Beside it was the best, we do not think that too much can be done for improvement of the approach. Only possibility is checking for more loss functions. However, we are skeptic that these results can be improved with a new loss function.

We have also mentioned generalized mean squared error, and we have seen its small application for a real data case. However, that application did not improve the result. For this approach we advise doing serious simulation study, together with improving the suboptimal λ. That can also be improved such that we estimate the $GMSE$ function, and then we optimize it.

At the end, as answers for our objectives we have:

- We have constructed estimators which improve MSE of the unbiased estimators. The improvements occur in so many cases that we may assume it holds in general.
- We did not improve the Brown's result. We did not even come close to it.

7 Povzetek magistrskega dela v slovenskem jeziku

Ena od glavnih lastnosti, ki jo preučujemo pri konstrukciji cenilk, je nepristranskost. Cenilka je nepristranska, ko je njena pričakovana vrednost enaka pravi vrednosti parametra, ki ga ocenjujemo. Druga pomembna lastnost, ki jo želimo, je ta, da ima cenilka čim manjšo varianco. Obstaja veliko načinov, kako najti nepristransko cenilko z najmanjšo možno varianco, za različne parametre iz različnih porazdelitev. V tem magistrskem delu pa nas zanima drugačen vidik. Vprašamo se, kaj se zgodi, ko pogoj o nepristranskosti cenilke izpustimo. Zanima nas, ali lahko konstruiramo cenilko, ki ni nepristranska, ampak ima manjšo varianco kot nepristranska cenilka. Mera, ki skupaj meri pristranskost in varianco cenilke, se imenuje srednja kvadratna napaka - SKN (ang. Mean squared error - MSE). Ko imamo to definirano, želimo poiskati cenilko, ki ima manjšo SKN kot nepristranska cenilka. Izkaže se, da v nekaterih primerih lahko poiščemo takšno cenilko. Prvi rezultat na tem področju je podal Stein v članku [16]. Stein je za lokacijski parameter multivariatne normalne porazdelitve podal pristransko cenilko, ki ima manjšo SKN, kot je povprečje podatkov, za katerega vemo, da je nepristranska cenilka. Rezultat je bil nepričakovan in zato se ta fenomen danes imenuje Steinov paradoks. Vprašanje pa je, ali se lahko enako naredi za Bernoullijevo porazdelitev. Takšen problem je Brown obravnaval v članku [4] na podatkih o uspešnosti odboja za igralce pri igri baseball. Ideja njegove rešitve je ta, da se binomsko porazdelitev, ki je vsota Bernoullijevih spremenljivk, za veliko število podatkov, lahko aproksimira z normalno porazdelitvijo, kar omogoča uporabo Steinove cenilke.

V magistrskem delu smo k opisanemu problemu pristopili drugače, brez uporabe normalne aproksimacije. Problem smo preučevali s teoretičnega in praktičnega vidika. Glavno orodje s katerim smo poizkušali rešiti problem, je model logistične regresije. Najprej smo naredili študijo problema napovedovanja z vidika statistične teorije učenja in o logistični regresiji v splošnem. V našem primeru logistično regresijo uporabljamo za primere, ko so vse neodvisne spremenljivke opisne. Ko imamo takšen primer, je pogojna porazdelitev odvisne spremenljivke Bernoullijeva. Ko smo izvedli logistično regresijo za tovrstne podatke, nam ocenjevanje parametrov z metodo največjega verjetja poda nepristransko cenilko za parameter Bernoullijeve porazdelitve. Tukaj pride na vrsto

penalizacija.

Penalizacija v napovedalnih modelih ima vlogo zmanjševanja preprileganja, vendar mi smo jo uporabili z namenom, da bi izboljšali nepristransko cenilko. Uporabili smo L2 penalizacijo (ang. ridge penalty). V splošnem se regresijski model skupaj z L2 penalizacijo imenuje L2 regresija (ang. ridge regression), v našem primeru tako lahko govorimo o L2 logistični regresiji. L2 regresija vsebuje penalizacijski koeficient, od vrednosti katerega bo odvisna tudi dobljena (penalizirana) cenilka. Cenilko z manjšo SKN smo poizkusili dobiti na ta način, da smo iskali optimalen penalizacijski koeficient za različne kriterijske funkcije. Kriterijska funkcija je lahko tista, ki se jo dobi z uporabo metode največjega verjetja ali pa lahko tudi kakšna druga. Možne kriterijske funkcije so opisane v [13]. Takšen optimizacijski problem je zahteven in njegovo reševanje predstavlja glavni teoretični prispevek tega magistrskega dela. Na to temo je bilo narejenega že veliko dela in to je opisano v neobjavljenih člankih, kjer so Blagus et al. pokazali obstoj takšnega penalizacijskega koeficienta za različne tipe L2 penalizacije, ampak niso podali načina, kako ga dejansko oceniti iz dejanskih podatkov [14], kar je temeljni prispevek tega magistrskega dela.

Po teoretični analizi, je na koncu magistrskega dela narejena še velika simulacijska študija, v kateri smo za velik nabor primerov poizkusili poiskati numerično optimalno rešitev.

Na koncu smo predlagane metode uporabili tudi na podatkih, ki jih je obravnaval že Brown. Pokazali smo, da lahko z uporabo predlaganih metod izboljšamo rezultate v primerjavi z standardno cenilko, žal pa nismo uspeli izboljšati rezultatov do katerih je prišel Brown z uporabo njegovega pristopa. Pridobljene rezultate smo v zaključku ovrednotili in podali možnosti za nadaljnje raziskovanje.

8 Bibliography

[1] P Bartlett, M Jordan, and J McAuliffe. Convexity, classification, and risk bounds (technical report 638). *Department of Statistics, UC Berkeley*, 2003. *(Cited on page 14.)*

[2] VP Bhapkar. Cramer-rao inequality. *Wiley StatsRef: Statistics Reference Online*, 2014. *(Cited on page 3.)*

[3] George EP Box, William Gordon Hunter, J Stuart Hunter, et al. Statistics for experimenters. 1978. *(Cited on page 6.)*

[4] Lawrence D Brown. In-season prediction of batting averages: A field test of empirical bayes and bayes methodologies. *The Annals of Applied Statistics*, pages 113–152, 2008. *(Cited on pages 5, 31, and 63.)*

[5] Patrick L Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011. *(Cited on page 42.)*

[6] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977. *(Cited on page 57.)*

[7] MA Girshick and LJ Savage. Bayes and minimax estimates for quadratic loss functions. Technical report, STANFORD UNIVERSITY STANFORD United States, 1951. *(Cited on page 3.)*

[8] Robby Haelterman. *Analytical study of the least squares quasi-Newton method for interaction problems*. PhD thesis, Ghent University, 2009. *(Cited on page 42.)*

[9] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. *(Cited on page 47.)*

[10] Krzysztof C Kiwiel. Convergence and efficiency of subgradient methods for quasiconvex minimization. *Mathematical programming*, 90(1):1–25, 2001. *(Cited on page 40.)*

[11] Saskia Le Cessie and Johannes C Van Houwelingen. Ridge estimators in logistic regression. *Applied statistics*, pages 191–201, 1992. *(Cited on page 22.)*

[12] Ronald L Plackett. Some theorems in least squares. *Biometrika*, 37(1/2):149–157, 1950. *(Cited on pages 3 and 21.)*

[13] Georg Heinze Rok Blagus, Hana ˇ Sinkovec. On the convergence of tuned logistic ridge regression under separation. unpublished, N.D. *(Cited on page 64.)*

[14] Jelle J. Goeman Rok Blagus. Mean squared error of ridge estimators in logistic regression. unpublished, N.D. *(Cited on pages 23, 33, and 64.)*

[15] Fadil Santosa and William W Symes. Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330, 1986. *(Cited on page 21.)*

[16] Charles Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. Technical report, STANFORD UNIVERSITY STANFORD United States, 1956. *(Cited on pages 3 and 63.)*

[17] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. *(Cited on page 21.)*

[18] G. van Rossum and F.L. Drake (eds). Python reference manual, 2001–. *(Cited on page 47.)*