Technische Universiteit Delft
Faculteit Elektrotechniek, Wiskunde en Informatica
Delft Institute of Applied Mathematics

# Statistical analysis of newspaper headlines with optimization

# Statistische analyse van krantenkoppen met optimalisatie

Verslag ten behoeve van het
Delft Institute for Applied Mathematics
als onderdeel ter verkrijging

van de graad van

**BACHELOR OF SCIENCE**
**in**
**TECHNISCHE WISKUNDE**

**door**

**ANNEKE JACOBS**

**Delft, Nederland**
**Augustus 2011**

**BSc verslag TECHNISCHE WISKUNDE**

# “Statistische analyse van krantenkoppen met optimalisatie”
# “Statistical analysis of newspaper headlines with optimization”

Anneke Jacobs

**Technische Universiteit Delft**

**Begeleider**

Dr. F. Vallentin

**Overige commissieleden**

| | |
|---|---|
| Dr. ir. M.C. Veraar | Dr. G.F. Ridderbos |

Augustus, 2011 Delft

# Preface

During my study I became more and more interested in the fields of Optimization and Statistics. While exploring my options for my bachelor thesis, I met dr. Frank Vallentin, who had a practical project about analyzing newspapers. At first glance I did not see how to use mathematics to analyze the news, but now I am finished I can tell you that there actually is a lot of math behind it! I have written this thesis as the final assignment for my bachelor's degree in Industrial and Applied Mathematics at Delft University of Technology.

I would like to thank Floris for believing in me and being there for me at any time when I needed to talk about the project. Also I want to thank my parents for all the support they have given me during my study. And at last, I want to thank my supervisor, dr. Frank Vallentin for his great help during this project.

Anneke Jacobs
Delft, August 2011

# Contents

# Introduction

What is going on in the world and in our country is primarily brought to us via news media. What we want to investigate is what the differences are between certain Dutch news papers such as 'NRC Handelsblad' and 'de Volkskrant' in how they portray the news. To investigate this, we use the $l_1$-regularized logistic regression method as described in the article 'Discovering word associations in news media via feature selection and sparse classification' [3].

The $l_1$-regularized logistic regression method generates a list of words which are strongly associated with the given query in the given dataset. The dataset can be a set of headlines, full articles or even only the first paragraph of an article. The program works as follows: Based on a given query, the dataset is separated into two classes consisting of items who do contain the query and those that do not. Applying the $l_1$-regularized logistic regression method gives a weight (positive or negative) to each word appearing in any of those two classes. A large positive weight means that that word has strong association with the query and a large negative weight means that there is not any association with the query. The weight vector is a sparse vector, which is a vector where most of the elements are equal to zero. An example of a result can be seen in Figure 1. The fontsize represents the extent to which the word is associated with the query, the colors of the words have no meaning.

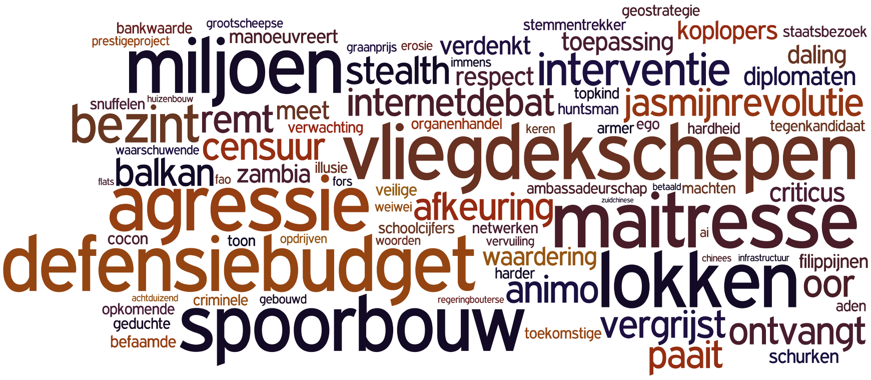


Figure 1: Wordcloud Dataset1, query `china`

The thesis can be divided in two parts: A theoretical part (chapters one to four) and a practical part (chapter five). The first chapter introduces the basic theory of convex sets and functions. Also the norm function will be described which is an important part of the $l_1$-regularized logistic regression method. In chapter two the logistic regression model is introduced. Logistic regression is used to make a prediction of the probability of the occurence of an event by fitting data to a logistic curve. This function is a non-linear function which makes it difficult to estimate the parameters. One way to solve this problem is to solve the corresponding convex optimization problem. For these problems Newton's method can be used to approximate an optimal solution, which is described in chapter three. In chapter four, the $l_1$-regularized logistic regression is introduced. This method is more general than the logistic regression method because one has the extra possibility to choose weights. Despite the benefits of introducing the extra term, the $l_1$-regularized logistic regression method is not able to use Newton's method due to the fact that the function to be optimized is no longer differentiable. However, there are methods in modern convex optimization (beyond the scope of this project) which can be applied here and which heavily rely on Newton's method. In chapter five a description is given of how data can be transformed into the right format for the software to use it and how the parameter $\lambda$ is found. For this part of the project I have written two Matlab codes, which can be found in appendix B.

The theory is largely based on the book 'Convex Optimization', written by Stephen Boyd and Lieven Vandenberghe [1]. Stephen Boyd is also one of the designers of the software used.

There are four datasets used for this project. The first three datasets are based on headlines from the section 'buitenland' from two different newspapers. Two datasets are from NRC Handelsblad and the other dataset is from de Volkskrant. The fourth dataset is a list of most common Dutch words which are substracted from the dataset. The datasets are stored online, they can be downloaded via the following links.

Dataset1 (NRC Handelsblad): `http://db.tt/qS0PHua`,
Dataset2 (NRC Handelsblad): `http://db.tt/O6QxG2B`,
Dataset3 (de Volkskrant): `http://db.tt/IluVj0U`,
Dataset4 (Most common words): `http://db.tt/eyRoAKP`.

Chapter 1

# Convex Sets and Functions

In this chapter the basic theory of convex sets and convex function is given. In the last part of this chapter the norm function and several properties are described. This chapter forms the theoretical basis for chapters two and three.

## 1.1 Convex sets and functions

A set $V \subseteq \mathbb{R}^n$ is convex if the line segment between any two points in $V$ lies in $V$. So, for any $x, y \in V$ and any $\theta$ with $0 \leq \theta \leq 1$ the following must hold

$$\theta x + (1-\theta)y \in V.$$

This means that the line segment connecting two arbitrary points of the set should stay in the set. For instance, cubes, circles or ellipsoids are convex sets where a star-shaped object is not. An example of a convex and a non-convex set in $\mathbb{R}^2$ can be seen in Figure 1.1.

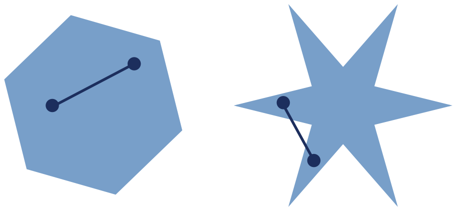


Figure 1.1: A convex set (left) and a non-convex set (right)

A function $f : \mathscr{D}(f) \to \mathbb{R}$ is convex if the domain of $f$, $\mathscr{D}(f) \subseteq \mathbb{R}^n$, is a convex set and if for all $x, y \in \mathscr{D}(f)$ and $0 \leq \theta \leq 1$

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y). \tag{1.1}$$

The function is strictly convex if strict inequality holds in expression (1.1) whenever $x \neq y$ and $0 < \theta < 1$. The function is concave if $-f$ is convex, and strictly concave if $-f$ is strictly convex.

Convex sets and a convex functions are related to each other, namely through the epigraph.

**Definition 1.1.** The epigraph of a function $f : \mathscr{D}(f) \to \mathbb{R}$ is defined as

$$epi(f) = \{(x,y)|x \in \mathscr{D}(f), f(x) \le y\}.$$

**Theorem 1.2.** *A function $f : \mathscr{D}(f) \to \mathbb{R}$ is a convex function if and only if $epi(f)$ is a convex set.*

*Proof.* Given any $(x_1,y_1),(x_2,y_2) \in epi(f)$ and $0 \le \theta \le 1$.

$\Rightarrow$: Now by the definition of a convex function

$$\begin{aligned} f(\theta x_1 + (1-\theta)x_2) &\le \theta f(x_1) + (1-\theta)f(x_2), \\ &\le \theta y_1 + (1-\theta)y_2. \end{aligned}$$

This last inequality follows from the definition of the epigraph and thereby the epigraph is a convex set.
$\Leftarrow$: Assume that $epi(f)$ is a convex set. Then,

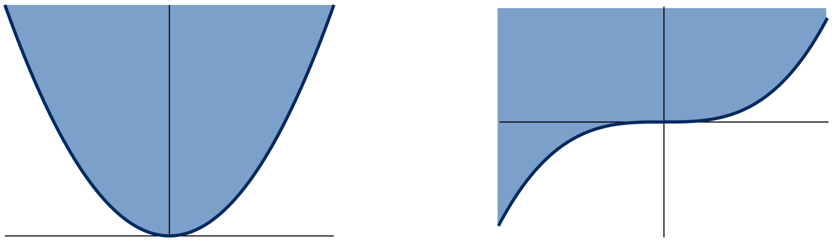
$$\theta(x_1, f(x_1)) + (1-\theta)(x_2, f(x_2)) = (\theta x_1 + (1-\theta)x_2, \theta f(x_1) + (1-\theta)f(x_2)) \in epi(f).$$

Then by the definition of the epigraph

$$f(\theta x_1 + (1-\theta)x_2) \le \theta f(x_1) + (1-\theta)f(x_2),$$

thus $f$ is a convex function. □

In subfigures (a) and (b) of Figure 1.2 two graphs with their epigraphs are shown. It can be easily calculated that $f(x) = x^2$ is a convex function on $\mathbb{R}$ and $f(x) = x^3$ is not. This can also be seen from the epigraphs of those functions. For each two points $(x_1,y_1)$ and $(x_2,y_2)$ from the epigraph of $f(x) = x^2$, the line segment connecting those points stays in the set. This is not the case for $f(x) = x^3$, take for example $(x_1,y_1) = (-1,-1)$ and $(x_2,y_2) = (0,0)$.



(a) $f = x^2$

(b) $f = x^3$

Figure 1.2: Epigraphs of a convex and a non-convex function

If a function $f$ has a continuous second derivative and $\mathscr{D}(f) \subseteq \mathbb{R}$, you can instead of using the definition also use the following theorem to see if a function is a convex function or not.

**Theorem 1.3.** *A twice differentiable function $f : \mathscr{D}(f) \to \mathbb{R}$ with a continuous second derivative and $\mathscr{D}(f) \subseteq \mathbb{R}$, is convex on the convex set $\mathscr{D}(f)$ if and only if $f''(x) \ge 0$ for all $x \in \mathscr{D}(f)$.*

*Proof.* Suppose $x, y \in \mathscr{D}(f)$, then by the mean value theorem there exists a $z$ with $x < z < y$ such that

$$\frac{f(y) - f(x)}{y - x} = f'(z). \tag{1.2}$$

Because $f''(x) \geq 0$, the function $f'(x)$ is monotonically increasing and hence $f'(x) \leq f'(z) \leq f'(y)$. Rewriting (1.2) gives

$$\begin{aligned} f(y) &= f(x) + f'(z)(y - x) \\ &\geq f(x) + f'(x)(y - x). \end{aligned}$$

In the same way one has

$$\begin{aligned} f(x) &= f(y) + f'(z)(x - y) \\ &\geq f(y) + f'(y)(x - y). \end{aligned}$$

Take $x \leq u \leq y$ with $u = \theta x + (1 - \theta)y$ and $0 \leq \theta \leq 1$, then

$$f(x) \geq f(u) + f'(u)(x - u) \tag{1.3}$$

$$f(y) \geq f(u) + f'(u)(y - u). \tag{1.4}$$

If (1.3) is multiplied by $\theta$ and (1.4) by $(1 - \theta)$ and adding up gives

$$\begin{aligned} \theta f(x) + (1 - \theta)f(y) &\geq \theta f(u) + \theta f'(u)(y - u) + (1 - \theta)f(u) + (1 - \theta)f'(u)(y - u) \\ &= f(u) + (\theta x + (1 - \theta)y - u)f'(u) \\ &= f(u) + (u - u)f'(u) \\ &= f(u) \\ &= f(\theta x + (1 - \theta)y). \end{aligned}$$

□

**Theorem 1.4.** *If $f, g : \mathscr{D}(f) \to \mathbb{R}$ are convex functions, then the sum of $f$ and $g$ is also a convex function.*

*Proof.* Suppose $h : \mathscr{D}(f) \to \mathbb{R}$ with $h = f + g$ and $f, g$ are convex functions. Then, for $x, y \in \mathscr{D}(f)$ and $0 \leq \theta \leq 1$,

$$\begin{aligned} h(\theta x + (1 - \theta)y) &= f(\theta x + (1 - \theta)y) + g(\theta x + (1 - \theta)y) \\ &\leq \theta f(x) + (1 - \theta)f(y) + \theta g(x) + (1 - \theta)g(y) \\ &= \theta(f(x) + g(x)) + (1 - \theta)(f(y) + g(y)) \\ &= \theta h(x) + (1 - \theta)h(y). \end{aligned}$$

□

## 1.2 Norm functions

A norm is a function that assigns a strictly positive length or size to all vectors in a vector space, other than the zero vector.

**Definition 1.5.** A function $||\cdot|| : \mathbb{R}^n \to \mathbb{R}$ is called a norm if

1. $||\cdot||$ is nonnegative: $||x|| \geq 0$ for all $x \in \mathbb{R}^n$,
2. $||\cdot||$ is definite: $||x|| = 0$ only if $x = 0$,
3. $||\cdot||$ is homogeneous: $||\lambda x|| = |\lambda|||x||$ for all $x \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$,
4. $||\cdot||$ satisfies the triangle inequality: $||x + y|| \leq ||x|| + ||y||$ for all $x, y \in \mathbb{R}^n$.

The most common norm is the $l_2$-norm. This norm gives the length of a vector $x = (x_1, \ldots, x_n)$ by

$$||x||_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}.$$

The $l_1$-norm is given by

$$||x||_1 = \sum_{i=1}^{n} |x_i|.$$

This norm is also called the taxicab norm. The name relates to the distance a taxi has to drive in a rectangular street grid to get from the origin to point $x$. This norm plays a significant role in the $l_1$-regularized logistic regression method.

The two given norms come from the general form of the $p$-norm. That is, let $p \geq 1$ be a real number, then the $l_p$-norm is given by

$$||x||_p = \left( \sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}.$$

An important property of the norm is that every norm is a convex function.

**Theorem 1.6.** *Every norm $||\cdot||$ is a convex function.*

*Proof.* Suppose $x, y \in \mathbb{R}^n$ and $0 \leq \theta \leq 1$ then, by the properties of the norm,

$$\begin{aligned} ||\theta x + (1-\theta)y|| &\leq ||\theta x|| + ||(1-\theta)y|| && \text{by property 4} \\ &= |\theta|||x|| + |(1-\theta)|||y|| && \text{by property 3} \\ &= \theta||x|| + (1-\theta)||y||, \end{aligned}$$

and the result follows. □

Chapter 2

# Logistic Regression

Logistic regression is a type of a predictive model that can be used when the response variable is a categorical variable with only two categories, for example failure and success. One of the questions one can ask is what the probability is that a person dies from a heart disease within a specified time period given some factors like gender, age and body mass index.

First, the logistic regression model is explained with only one explanatory variable and shall later on be extended to a multiple regression model with several explanatory variables. Secondly, given a datamatrix of explanatory variables with known outcomes, the coefficients $\beta_i$ can be found via maximum likelikehood estimates. The theory follows the book 'Applied Regression analysis and generalized linear models' from John Fox [2].

## 2.1 Single regression

Suppose you want to know what the probability is of dying from a heart disease within the next ten years depending on the age you are. The response variable $Y$ can be seen as follows

$$Y = \begin{cases} 1 & \text{you die from a heart disease within the next ten years,} \\ -1 & \text{you will not die from a heart disease within the next ten years,} \end{cases}$$

and $X$ is your age in years. A scatter plot of this example is depicted in Figure 2.1. The points are jittered horizontally and vertically to minimize overplotting. This dataset comes from the Evans County study [9]. The study was a cohort study of men followed for seven years, there were 609 participiants in the study. The conditional probability $\pi_i$ is given by

$$\pi_i \equiv \Pr(Y_i) \equiv \Pr(Y = 1|X = x_i).$$

The other outcome has probability $\Pr(Y = -1|X = x_i) = 1 - \pi_i$. The expectation can be easily calculated:

$$E(Y|x_i) = \pi_i \cdot (1) + (1 - \pi_i) \cdot (-1) = 2\pi_i - 1.$$

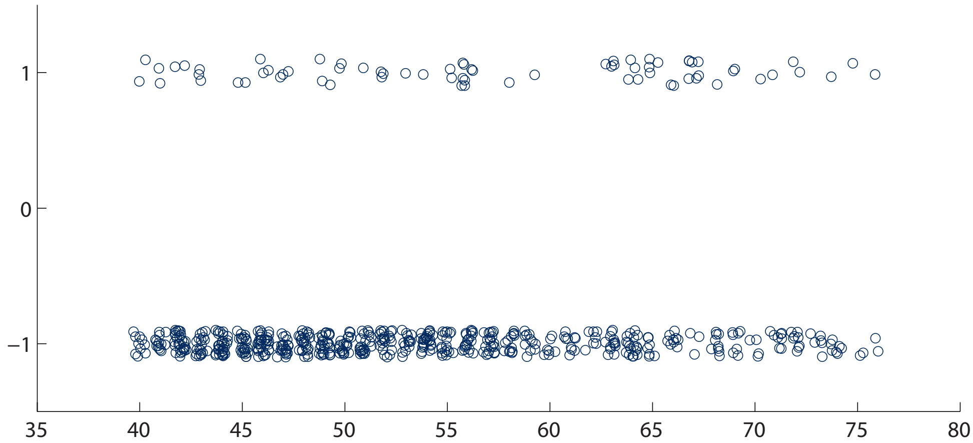


Figure 2.1: Jittered scatterplot of age $X_i$ versus outcome $Y_i$

What we want to find is a relationship between the conditional probabilty and the variable $X$.

### 2.1.1 Transformation of $\pi$

The conditional probability cannot be modeled by a linear predictor $\eta = \beta_0 + \beta_1 X$, because of the possibility that $\eta$ have values outside the unit interval. A positive, monotone function $P(\cdot)$ which maps the linear predictor into the unit interval can resolve this problem. So the transformation looks like

$$\pi_i = P(\eta_i) = P(\beta_0 + \beta_1 X_i),$$

and $\beta_0$ and $\beta_1$ are the parameters to be estimated. The function $P(\cdot)$ should be both smooth and symmetric and should approach $\pi = 0$ and $\pi = 1$ as asymptotes.

The logistic distribution function is defined as

$$\Lambda(z) = \frac{1}{1 + \exp(-z)},$$

and produces the linear logistic regression model:

$$\begin{aligned}\pi_i &= \Lambda(\beta_0 + \beta_1 X_i) \\ &= \frac{1}{1 + \exp[-(\beta_0 + \beta_1 X_i)]}.\end{aligned}$$

The inverse of the logistic distribution function is also called the logit function and is equal to

$$\Lambda^{-1}(\pi_i) = \ln\left(\frac{\pi_i}{1 - \pi_i}\right) = \beta_0 + \beta_1 X_i.$$

The term $\frac{\pi_i}{1-\pi_i}$ is called the odds. Because the probabilities lie between 0 and 1, odds lie between zero and infinity. The logit function is an odd function and has $\pi = 0$ and $\pi = 1$ as asymptotes. The linear predictor $\eta$ can take values from minus infinity to infinity. When the probability of the two outcomes are equal, they have both probability 0.5, the logit function is equal to 0.

## 2.2 Multiple regression

Generalizing the logistic distribution model to several explanatory variables is straightforward. The only difference is that the linear predictor $\eta$ is now a function of several explanatory variables. This means that

$$\begin{aligned}\pi_i &= \Lambda(\eta_i) = \Lambda(\beta_0 + \beta_1 X_{1,i} + \cdots + \beta_k X_{k,i}) \\ &= \frac{1}{1+\exp[-(\beta_0 + \beta_1 X_{1,i} + \cdots + \beta_k X_{k,i})]}.\end{aligned} \tag{2.1}$$

Equation (2.1) can be rewritten to

$$\ln\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 X_{1,i} + \cdots + \beta_k X_{k,i}.$$

The relationship between the linear predictor $\eta$ and the conditional probability $\pi$ is depicted in Figure 2.2.

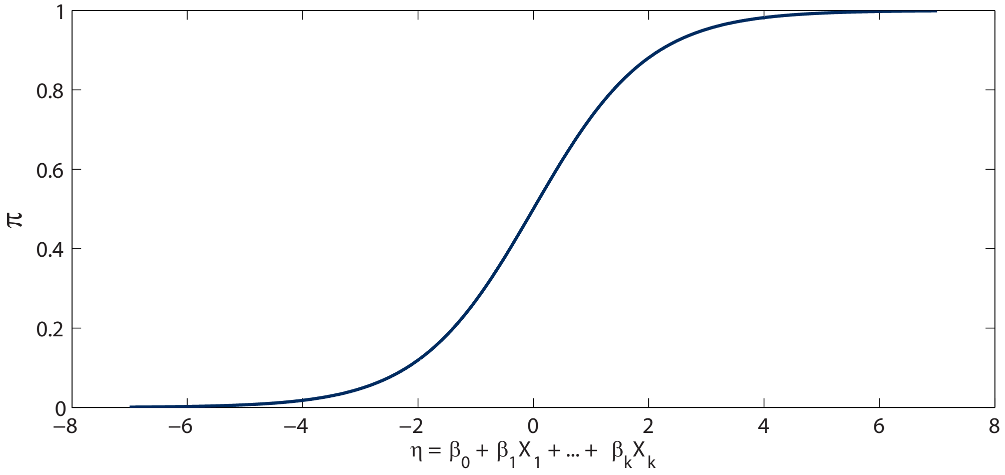


Figure 2.2: Relationship between linear predictor $\eta$ and conditional probability $\pi$

Returning to the example used in the first part, what is the probability that you die from a heart disease within the next ten years? In this model [7], there are six explanatory variables to predict risk of death from heart disease in the next ten years:

$X_1 =$ Smoking, where 0 is for non-smoking and 1 is smoking,

$X_2 =$ Total Cholesterol Level (TCL - 200),

$X_3 =$ Body Mass Index (BMI - 25),

$X_4 =$ Gender, where 0 is female and 1 is male,

$X_5 =$ Age, (years - 50),

$X_6 =$ Hours of physical activity per week.

After fitting the data, the $\beta_i$'s are estimated, which results in the following equation for the linear predictor:

$$\eta_i = -4.123 + 0.898X_{1,i} + 0.166X_{2,i} + 0.058X_{3,i} + 0.028X_{4,i} + 0.024X_{5,i} - 1.013X_{6,i}$$

A 55-year old woman, who smokes and has a TCL of 230 and a BMI of 32 and is physically inactive has an 86% chance of dying of a heart disease within the next ten years:

$$\begin{aligned}\eta &= -4.123 + 0.898 \cdot 1 + 0.166 \cdot 30 + 0.058 \cdot 7 + 0.028 \cdot 0 + 0.024 \cdot 5 - 1.013 \cdot 0 = 1.803546,\\ \pi_i &= \frac{1}{1+\exp(-1.803546)} = 0.859.\end{aligned}$$

On the contrary, a health-conscience 65 year old man who does not smoke, has a TCL of 180, BMI of 25 and sports 4 hours a week has almost zero chance of dying of a heart disease within the next ten years.

$$\begin{aligned}\eta &= -4.123 + 0.898 \cdot 0 + 0.166 \cdot -20 + 0.058 \cdot 0 + 0.028 \cdot 1 + 0.024 \cdot 15 - 1.013 \cdot 4 = -11.107,\\ \pi_i &= \frac{1}{1+\exp(11.107)} = 1.5 \cdot 10^{-5}.\end{aligned}$$

## 2.3 Estimating $\beta_i$

Until now the coefficients $\beta_i$ were given without any justification. Consider a sample of $n$ independent obervations and $k$ explanatory variables. Then there is an $n \times 1$ outcome vector $\mathbf{Y}$ and an $n \times k$ data matrix $\mathbf{X}$. The coefficients $\beta_i$ can be found by the maximum likelihood estimator. Because the observations are independent, the likelihood function looks like $\prod_{i=1}^{n} \pi_i$. When taking logarithms, the log-likelihood function is found:

$$\begin{aligned} l(\beta_0, \ldots, \beta_k) &= \log\left(\prod_{i=1}^{n} \pi_i\right)\\ &= \sum_{i=1}^{n} \log(\pi_i)\\ &= \sum_{i=1}^{n} \log\left(\frac{1}{1+\exp[-(\beta_0 + \beta_1 X_{1,i} + \cdots + \beta_k X_{k,i})]}\right)\\ &= -\sum_{i=1}^{n} \log\left(1+\exp[-(\beta_0 + \beta_1 X_{1,i} + \cdots + \beta_k X_{k,i})]\right)\\ &= -\sum_{i=1}^{n} f(\beta_0 + \beta_1 X_{1,i} + \cdots + \beta_k X_{k,i}),\end{aligned}$$

where $f$ is the logistic loss function:

$$f(z) = \log(1+\exp(-z)). \tag{2.2}$$

**Theorem 2.1.** *The logistic loss function $f(z) = \log(1+\exp(-z))$ is a convex function.*

*Proof.* Use theorem 1.3: The first and second derivatives of equation (2.2) are

$$\begin{aligned}
f'(z) &= \frac{1}{1+\exp(-z)} \cdot (-\exp(-z)) \\
&= -\frac{\exp(-z)}{1+\exp(-z)} \\
f''(z) &= -\frac{(1+\exp(-z))\cdot(-\exp(-z)) - \exp(-z)\cdot(-\exp(-z))}{(1+\exp(-z))^2} \\
&= \frac{\exp(-z)+\exp(-2z)-\exp(-2z)}{(1+\exp(-z))^2} \\
&= \frac{\exp(-z)}{(1+\exp(-z))^2}.
\end{aligned}$$

The last equation is always greater or equal to zero because of the exponential function in the numerator and the square in the denominator which are both always positive. □

Because the logistic loss function is a convex function, the log-likelihood function is concave. The negative of the log-likelihood function is called the (emperical) logistic loss. If the logistic loss function is divided by $n$, the average logistic loss is found

$$\mathscr{L}(\beta_0,\dots,\beta_k) = \frac{1}{n}\sum_{i=1}^{n} f(\beta_0+\beta_1 X_{1,i}+\cdots+\beta_k X_{k,i}).$$

Since $\mathscr{L}(\beta_0,\dots,\beta_k)$ is a convex function of the parameters $\beta_i$, the logistic regression problem can be solved by solving the convex optimization problem

$$\text{minimize} \quad \mathscr{L}(\beta_0,\dots,\beta_k), \tag{2.3}$$

with parameters $\beta_i \in \mathbb{R}$, data matrix $\mathbf{X}$ and outcome vector $\mathbf{Y}$. From theorem 3.2 it follows that the maximum likelihood estimates for $\beta_i$ can be found by solving the system of $k+1$ simultaneous equations

$$\frac{\partial \mathscr{L}}{\partial \beta_i} = 0 \qquad \text{for } i = 0,\dots,k.$$

With Newton's method an approximation of the solution can be found. This method will be explained in the following chapter.

Chapter 3

# Convex Optimization

In this chapter the theory of optimization is described. Starting with the standard form of an optimization problem, the basic terms and notations are described. Later on the convex optimization problem is described, which is a special case of a standard optimization problem. In the last part Newton's method is described which can solve unconstrained convex optimization problems.

The standard form of a optimization problem looks like

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0 & i = 1, \dots, m \\ & h_j(x) = 0 & j = 1, \dots, p. \end{array}$$

Solving this problem is finding an $x$ which minimizes $f_0(x)$ and satisfies the conditions $f_i(x) \leq 0$ and $h_j(x) = 0$. In this problem $x \in \mathbb{R}^n$ is the optimization variable and the function $f_0(x) : \mathbb{R}^n \to \mathbb{R}$ is called the objective function. The inequalities $f_i(x) \leq 0$ are the inequality constraints and the equations $h_j(x) = 0$ are the equality constraints.

The set of points for which the objective function and the constraint functions are defined

$$V = \bigcap_{i=0}^{m} \mathscr{D}(f_i) \cap \bigcap_{j=1}^{p} \mathscr{D}(h_j),$$

is called the domain of the optimization problem. A point $x \in V$ is feasible if it satisfies all the constraint functions. The optimal value $p^*$ of the problem is defined as

$$p^* = \inf \left\{ f_0(x) \mid f_i(x) \leq 0, i = 1, \dots, m,\ h_j(x) = 0, j = 1, \dots, p \right\}.$$

An optimal point $x^*$ solves the problem if $x^*$ is feasible and $f_0(x^*) = p^*$.

## 3.1 Convex optimization

The difference between a optimization problem and a convex optimization problem is that the objective function and the inequality constraint functions must be convex functions. Also, the equality constraint function $h_j(x) = a_j^T x - b_j$ must be affine (linear). The convex optimization problem looks thus as follows

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0 & i = 1, \dots, m \\ & a_j^T x = b_j & j = 1, \dots, p. \end{array}$$

Because of the convexity of the objective function any locally optimal point is also (globally) optimal.

**Theorem 3.1.** *If $f_0 : \mathscr{D}(f_0) \rightarrow \mathbb{R}$ is a convex function, then every local optimum is a global optimum.*

*Proof.* Suppose $x \in \mathscr{D}(f_0)$ is a local optimum, $x$ is feasible and

$$f_0(x) = \inf \{ f_0(z) \mid z \text{ feasible, } \|z - x\|_2 \leq R \}, \tag{3.1}$$

for some $R > 0$. Now suppose that $x$ is not a global optimum, that is, there is a $y$ such that $f_0(y) < f_0(x)$. By equation (3.1), $\|y - x\|_2 > R$. Suppose the point $z$ is given by

$$z = (1 - \theta)x + \theta y, \quad \text{with} \quad \theta = \frac{R}{2\|y - x\|_2}. \tag{3.2}$$

By defining $\theta$ in this way, $0 < \theta < \frac{1}{2}$. Combining equation (3.2) with equation (3.1) gives

$$\begin{aligned} \|z - x\|_2 &= \|(1 - \theta)x + \theta y - x\|_2 \\ &= \|\theta(y - x)\|_2 \\ &= \theta \|y - x\|_2 \\ &= \frac{R}{2\|y - x\|_2} \|y - x\|_2 = \frac{R}{2}. \end{aligned}$$

So $z$ is a feasible point by (3.1). Because $f_0$ is a convex function,

$$\begin{aligned} f_0(z) &\leq (1 - \theta) f_0(x) + \theta f_0(y) \\ &= f_0(x) + \theta(f_0(y) - f_0(x)) \\ &< f_0(x) + \frac{1}{2}(f_0(y) - f_0(x)) && \text{by (3.2)} \\ &= \frac{1}{2}(f_0(x) + f_0(y)) < f_0(x). \end{aligned}$$

The last expression follows by the definition of the global optimum $y$. This result contradicts with (3.1), therefore there exists no feasible $y$ with $f_0(y) < f_0(x)$, so $x$ is globally optimal. □

## 3.2 Newton's method for unconstrained minimization

In this section Newton's method is described which can solve the unconstrained convex optimization problem

$$\text{minimize} \qquad f(x), \tag{3.3}$$

with $f$ differentiable. Because $f$ is a convex function, the solution $x^*$ is found when

$$\nabla f(x^*) = 0, \tag{3.4}$$

where the gradient is given by

$$\nabla f(x) = \left(\frac{\partial}{\partial x_1} f(x), \ldots, \frac{\partial}{\partial x_n} f(x)\right)^T .$$

**Theorem 3.2.** *Suppose $f$ is differentiable and convex, then $x$ is optimal if and only if*

$$\nabla f(x) = 0.$$

*Proof.* Because $f$ is a differentiable and convex function, it follows from the definition of a convex function that for all $x, y \in \mathscr{D}(f)$

$$f(y) \geq f(x) + \nabla f(x)^T (y - x). \tag{3.5}$$

Proof of this statement can be found on page 70 of [1]. The point $x$ is optimal if for all feasible $y$ the following inequality holds

$$\nabla f(x)^T (y - x) \geq 0. \tag{3.6}$$

$\Rightarrow$: Suppose $x$ is optimal. Since $f$ is differentiable, its domain is open, so all $y$ sufficiently close to $x$ are feasible. Let $y = x - t\nabla f(x)$, with $t > 0$. If $t$ is chosen small, $y$ is feasible and

$$\begin{aligned}
\nabla f(x)^T (y - x) &= \nabla f(x)^T (x - t\nabla f(x) - x) \\
&= \nabla f(x)^T (-t\nabla f(x)) \\
&= -t||\nabla f(x)||_2^2 \geq 0.
\end{aligned}$$

The last equation only holds if $\nabla f(x) = 0$.
$\Leftarrow$: Suppose $\nabla f(x) = 0$ and $x$ satisfies (3.6), then by (3.5) $x$ is optimal. □

Solving equation (3.4) for $x^*$ is the same as solving the system of $n$ equations obtained by setting the $n$ partial derivatives equal to zero. In most of the cases these equations can not be solved analytically, but with Newton's method a solution can be approximated.

Netwon's method is an iterative algorithm which produces a minimizing sequence $x^{(k)}$, with $k = 1, 2, \ldots$ where

$$x^{(k+1)} = x^{(k)} + t^{(k)} \Delta x^{(k)},$$

where $t^{(k)} > 0$ unless $x^{(k)}$ is optimal. The vector $\Delta x$ is called the search direction and the scalar $t^{(k)} \geq 0$ is called the stepsize at iteration $k$. The method is a descent method, that is $f(x^{(k+1)}) < f(x^{(k)})$, except when $x^{(k)}$ is optimal. The search direction must satisfy

$$\nabla f(x^{(k)})^T \Delta x^{(k)} < 0. \tag{3.7}$$

The second-order Taylor approximation $T$ of $f$ around the point $a$ is

$$T(x+a) = f(a) + \nabla f(a)^T x + \frac{1}{2} x^T \nabla^2 f(a) x,$$

where the Hessian matrix of $f$ at $a$ is given by

$$\left[\nabla^2 f(a)\right]_{ij} = \frac{\partial^2}{\partial x_i \partial x_j} f(a).$$

This approximation of $f$ is a strictly convex quadratic function of $x$, so it has a unique minimizer $x^* \in \mathbb{R}^n$. This minimizer can be found by setting the gradient of $T$ equal to zero:

$$\begin{aligned} 0 &= \nabla T(x^*) \\ &= \left( \frac{\partial}{\partial x_1} T(x^*), \ldots, \frac{\partial}{\partial x_n} T(x^*) \right)^T \\ &= \nabla f(a) + \nabla^2 f(a) x^*. \end{aligned}$$

Solving for $x^*$ gives the Newton step for $f$ at $x$:

$$x^* = -(\nabla^2 f(a))^{-1} \nabla f(a) = \Delta x_{nt}.$$

Because $f$ is a convex function, $\nabla^2 f(x)$ is positive semidefinite. This can be proven like in theorem 1.3 which is the one-dimensional case of this statement. Equation (3.7) can be rewritten as

$$\nabla f(x)^T \Delta x_{nt} = -\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x) < 0,$$

unless $\nabla f(x) = 0$, so the Newton step is a descent direction (unless $x$ is optimal). Newton's method can now be stated

$$x^{(k+1)} = x^{(k)} + t^{(k)} \Delta x_{nt}^{(k)}.$$

Algorithm 2 represents the scheme to be followed. At each iteration $k$, the stepsize $t$ should be chosen such that $x^{(k+1)} \in \mathscr{D}(f)$, this can be done with backtracking line search (Algorithm 1). This algorithm starts with $t = 1$ and checks if the point $x^{(k+1)}$ is in the domain or not. If it is not, it sets $t = \frac{t}{2}$ and checks again and repeats this method until the first time that $x^{(k+1)} \in \mathscr{D}(f)$.

**Algorithm 1** Backtracking line search

```
- Given a descent direction $\Delta x_{nt}$ for $f$ at $x \in \mathscr{D}(f)$
- Set $t := 1$
**while** $x + t\Delta x_{nt} \notin \mathscr{D}(f)$ **do**
    - Set $t := \frac{1}{2}t$
**end while**
```

**Algorithm 2** Newton's method

```
Given a starting point $x \in \mathscr{D}(f)$ and tolerance $\varepsilon > 0$
**repeat**
    - Compute the Newton step
            $\Delta x_{nt} := -\nabla^2 f(x)^{-1} \nabla f(x)$
    - Choose stepsize $t$ by backtracking line search such that
            $x + t\Delta x_{nt} \in \mathscr{D}(f)$
    - Set
            $x := x + t\Delta x_{nt}$
**until** $\|\nabla f(x)\|_2 \leq \varepsilon$
```

Chapter 4

# $l_1$-Regularized Logistic Regression

This chapter describes the theory used in the `l1_logreg` software [6]. The article 'An interior point method for large-scale $l_1$-regularized logistic regression' [5] is the basis of the theory this chapter.

When $m$ is much smaller than $n$, logistic regression leads to over-fit. A technique to prevent over-fitting is regularization which is adding an extra term, the $l_1$-norm, which encourages the sum of the absolute values of the parameters to be small. The $l_1$-regularized logistic regression problem looks like

$$\text{minimize} \quad \mathscr{L}(\beta_0, \dots, \beta_k) + \lambda \|(\beta_1, \dots, \beta_k)\|_1$$

Or, equivalently

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} f(\beta_0 + \beta_1 X_{1,i} + \cdots + \beta_k X_{k,i}) + \lambda \sum_{i=1}^{k} |\beta_i| \tag{4.1}$$

where $\lambda > 0$ is the regularization parameter. This method is called the $l_1$-regularized logistic regression problem because of the introduction of the $l_1$-norm in the objective function. The objective function is a convex function but is not differentiable because of that norm. The convexity follows from combining theorems 1.3, 1.4 and 1.6.

The major advantage to use this method is that $l_1$-regularized logistic regression typically yields a sparse vector $(\beta_1, \dots, \beta_k)$. This means that most of the elements $\beta_i$ are equal to zero. The problem can be solved if the problem is transformed into one which has differentiable objective and constraint functions. So the $l_1$-regularized logistic regression problem can be solved by solving

$$\begin{array}{rl} \text{minimize} & \mathscr{L}(\beta_0, \dots, \beta_k) + \lambda 1^T u \\ \text{subject to} & -u_i \le \beta_i \le u_i, \end{array}$$

where $\mathbf{1}$ denotes the vector with all components equal to one and $u \in \mathbb{R}^n$. This new formulated problem is a convex optimization problem, with a smooth objective and constraint functions, so it can be solved by standard convex optimization methods such as the interior point algorithm which heavily rely on Newton's method. This theory is for instance explained in chapter 11 of [1] and in chapter 2 of [5], but goes beyond the scope of the project.

Chapter 5

# Application

In this chapter the theory of the previous chapters is put into practice. Newspapers store all their publications online and that was a good place to start. I have chosen two newpapers who are opposite in the political spectrum in the Netherlands. The first newspaper I have chosen is NRC Handelsblad, which is a right-wing newspaper. The other newspaper I used is de Volkskrant, a left-wing newspaper. It will be interesting to see if this difference will be reflected in the results.

The data is downloaded from the LexisNexis* website [8]. This website gathers all kind of publications including newspapers from the Netherlands. All datasets consists of headlines from section 'buitenland'. In this case I have used headlines, but is also possible to use the whole text of the article or just the first paragraph. Dataset1 contains all the headlines from October 13th, 2009 until June 23rd, 2011 from NRC Handelsblad. Dataset2 is also from NRC Handelsblad but is only from January 1st, 2011 until June 23rd, 2011. The last dataset I have used is Dataset3, which comes from de Volkskrant and is also from January 1st, 2011 until June 23rd, 2011.

When the data is downloaded from the internet, the dataset is just a large file with a lot of words in it and is not yet usable. The input for the `l1_logreg_train` method [6] should be a matrix, so a transformation of the data is needed. How to do this is described in the first section. After the data is in the right format, the actual application can begin. Choosing a query, we need to find a $\lambda$ which fits the data best. When the right $\lambda$ is found, two datasets are compared to each other to see if there is a difference between the two newspapers.

## 5.1 Transforming data

To show how the transformation of the data works, I have chosen to use a simple example of only four items. These items are based on headlines from the newspaper NRC Handelsblad.

```
'Kluizenaar' die wilde inbreken in de website van de CIA
Obama botst met Congres over legitimiteit missie Libië
Republikeinen willen zorgwet Obama afschaffen
VN: uitstel van 'genocide'-rapport
```

In appendix B.1, the Matlab code can be found that will transform the raw data into a suitable format. What this program basically does is the following:

1. Set each item on a separate line,
2. Convert all letters to lowercase,
3. Remove punctuation,
4. Remove diacritics. That is, for example, replacing `ë` by `e`.

After the transformation the data looks as follows:

```
kluizenaar die wilde inbreken in de website van de cia
obama botst met congres over legitimiteit missie libie
republikeinen willen zorgwet obama afschaffen
vn uitstel van genociderapport
```

In the text there are a lot of words such as adverbs and articles which occur frequently, but have almost no relation to the query, so they should be removed from the dataset. The 'Instituut voor Nederlandse Lexicologie' [4] has analyzed all the papers of the NRC Handelsblad in the years 1994 and 1995 and made a frequency list of all the words which appeared on one of the papers. A selection of the top 300 of these words is used which are discarded from the dataset.

The data is now almost in the right format. It need only be transformed into a matrix $M$. The size of matrix $M$ is $m \times n$, where $m$ is the number of items and $n$ the number of distinct features in the dataset. The elements of the matrix represents how many times a feature is in an item. The Matlab code for this part can be found in the second cell of the code in appendix B.2.

Continuing with the same example as before, this results in the following matrix:

$$M = \begin{array}{cccccccccccccccc}
\text{afschaffen} & \text{botst} & \text{cia} & \text{congres} & \text{genociderapport} & \text{inbreken} & \text{kluizenaar} & \text{legitimiteit} & \text{libie} & \text{missie} & \text{obama} & \text{republikeinen} & \text{uitstel} & \text{vn} & \text{website} & \text{zorgwet} \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0
\end{array}$$

As expected, $m = 4$ and in this case there are $n = 16$ features.

## 5.2 Query

A query is the word that we want to investigate. The query must be one of the features. After the query is chosen, the corresponding column is removed from matrix $M$ and a $m \times 1$ vector $N$ is made.

This vector has only two different elements, namely -1 and 1. If the queryword appears in item $i$, then $N(i)$ is equal to 1, if it does not, it is equal to -1. So, by example, if the queryword is `obama`, then the vector $N$ equals

$$N = \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix}.$$

With the transformed matrix $M$ and the vector $N$ the data is now ready for use for the `l1_logreg_train` method.

## 5.3 Variation of parameter $\lambda$

The parameter $\lambda$, as introduced in equation (4.1), roughly controls the number of nonzero components. For each dataset I have varied $\lambda$ from 1 to 0.000001 for the query `obama`, the results of these simulations can be found in appendix A. The tables consist of four columns. The first two columns speak for themselves, the third and fourth column represent the number of features in the positive or negative cloud. If a feature has a positive weight, and is therefore in the positive cloud, there is a strong association with the given query. If the feature is in the negative cloud, there is strong evidence that there is no association with the query.

For almost all the simulations the number of features in the result increases or remains on the same level as $\lambda$ decreases. For all three datasets, there is no result for $\lambda > 0.01$. The results of Dataset2 and Dataset3 are what you would expect: Decreasing $\lambda$ increases the number of features found, until some value of $\lambda$ from where the number of features stays the same. Dataset1 on the other hand, has a sudden decrease of features in the set at $\lambda = 0.00001$. Also, from that $\lambda$ on the number of features in the negative cloud increases very fast. By trial and error the desired result can be found.

## 5.4 Comparison of two datasets

Dataset2 and Dataset3 both consists of headlines from January 1st 2011 until June 23rd 2011 from section 'buitenland', so it make sense to compare these two with each other. For a graphical presentation of the result, I have made wordclouds with Wordle[1]. In this way it can be easily seen which words have a strong association with the query: The colors of the words have no meaning, only the size of the font matters. The wordclouds used are positive clouds: So only the features with a positive weight coefficient are used.

In Figures 5.1 and 5.2 two wordclouds are shown of Dataset2 and Dataset3 respectively. The first thing you notice is that Dataset2 has relatively many words with the same large font, so their 'top' of words is a relatively large group where the features have almost the same weight. Dataset3 has only a few words with a large font. Another thing what stands out is that the result of Dataset2 contains a lot of words especially about the politics and economy of America such as `bezuinigingsplan`,

[1] http://www.wordle.net/

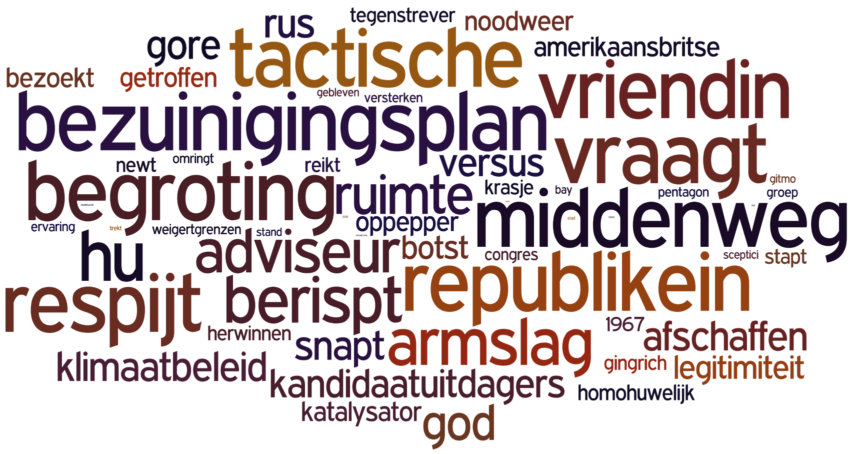


Figure 5.1: Wordcloud Dataset2 (NRC Handelsblad), query `obama`, $\lambda = 10^{-4}$

`republikein`, `begroting` and `klimaatbeleid`. Dataset3 on the other hand has more news from outside of America such as (`moosten` ('midden oosten'), `latijnsamerika`, `netanyahu` and `wereldmacht`. There are also more humanitarian words on personal level and on social issues such as `liefst`, `nare`, `gevoelige`, `zelfbeschikking`, `amicale`, `emotionele` and `zorgplan`. It is also notable that `barack` only appears in the result of Dataset3 and not in Dataset2.

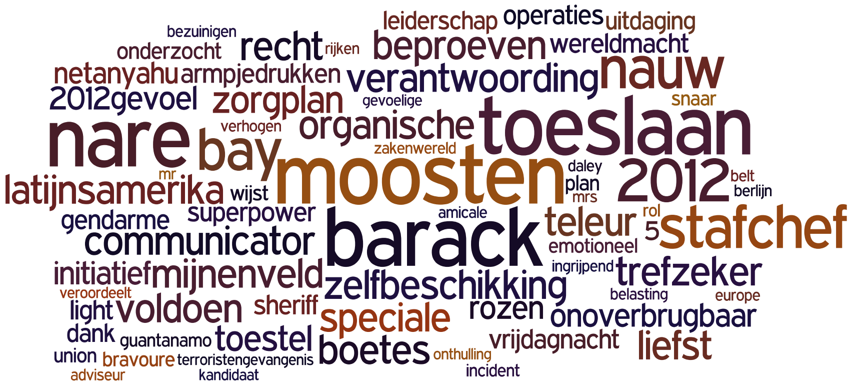


Figure 5.2: Wordcloud Dataset3 (de Volkskrant), query `obama`, $\lambda = 10^{-4}$

## 5.5 Conclusions and Recommendations

From the figures you can conclude that Dataset2 (NRC Handelsblad) has a more right conservative character and Dataset3 (de Volkskrant) tends more to the people. This is also what I expected.

For further research the following things can be considered: The datasets used are relatively small, a similiar research can be done with a much larger dataset, for example a datasets which contains all the headlines of the past ten years. For this project only headlines are used, but is also possible to extend it to text of the whole article. A flaw in the program is that the plural of nouns and verbs are seen as different words, as well as words like `den haag` are seen as two different words.

Appendix A

# Datasets

### Dataset1

Paper: NRC Handelsblad
Section: 'buitenland'
Type: headlines
Start date: October 13th, 2009
End date: June 23rd, 2011
Number of items: 5344
Number of features: 10943
Total calculation time: 733.92 sec
URL: `http://db.tt/qS0PHua`

| $\lambda$ | **time (sec)** | **positive cloud** | **negative cloud** |
|---|---|---|---|
| 1 | 1.72 | 0 | 0 |
| 0.5 | 1.65 | 0 | 0 |
| 0.1 | 1.70 | 0 | 0 |
| 0.05 | 1.71 | 0 | 0 |
| 0.01 | 2.26 | 144 | 0 |
| 0.005 | 1.76 | 179 | 0 |
| 0.001 | 2.31 | 208 | 79 |
| 0.0005 | 2.44 | 212 | 129 |
| 0.0001 | 3.31 | 212 | 197 |
| 0.00005 | 3.63 | 204 | 245 |
| 0.00001 | 2.09 | 44 | 54 |
| 0.000005 | 2.48 | 92 | 67 |
| 0.000001 | 5.09 | 194 | 2939 |

Table A.1: Dataset1, query `obama`. Positive/negative cloud: number of features with a positive/negative weight

### Dataset2

Paper: NRC Handelsblad
Section: ’buitenland’
Type: headlines
Start date: January 1st, 2011
End date: June 23rd, 2011
Number of items: 1407
Number of features: 4326
Total calculation time: 76.08 sec
URL: `http://db.tt/O6QxG2B`

| $\lambda$ | **time (sec)** | **positive cloud** | **negative cloud** |
|---|---|---|---|
| 1 | 0.42 | 0 | 0 |
| 0.5 | 0.38 | 0 | 0 |
| 0.1 | 0.37 | 0 | 0 |
| 0.05 | 0.79 | 0 | 0 |
| 0.01 | 0.54 | 60 | 0 |
| 0.005 | 0.53 | 61 | 0 |
| 0.001 | 0.54 | 62 | 8 |
| 0.0005 | 0.55 | 62 | 8 |
| 0.0001 | 0.59 | 62 | 8 |
| 0.00005 | 0.65 | 62 | 8 |
| 0.00001 | 0.53 | 62 | 8 |
| 0.000005 | 0.59 | 62 | 8 |
| 0.000001 | 0.54 | 57 | 8 |

Table A.2: Dataset2, query `obama`. Positive/negative cloud: number of features with a positive/negative weight

### Dataset3

Paper: de Volkskrant
Section: ’buitenland’
Type: headlines
Start date: January 1st, 2011
End date: June 23rd, 2011
Number of items: 1713
Number of features: 4517
Total calculation time: 95.80 sec
URL: `http://db.tt/IluVjOU`

| $\lambda$ | **time (sec)** | **positive cloud** | **negative cloud** |
|---|---|---|---|
| 1 | 0.45 | 0 | 0 |
| 0.5 | 0.40 | 0 | 0 |
| 0.1 | 0.40 | 0 | 0 |
| 0.05 | 0.41 | 0 | 0 |
| 0.01 | 0.59 | 69 | 0 |
| 0.005 | 0.56 | 71 | 1 |
| 0.001 | 0.57 | 71 | 6 |
| 0.0005 | 0.58 | 71 | 6 |
| 0.0001 | 0.57 | 71 | 20 |
| 0.00005 | 0.55 | 71 | 20 |
| 0.00001 | 0.56 | 71 | 20 |
| 0.000005 | 0.56 | 71 | 20 |
| 0.000001 | 0.53 | 71 | 20 |

Table A.3: Dataset3, query `obama`. Positive/negative cloud: number of features with a positive/negative weight

Appendix B

# Matlab code

## B.1 source2input.m

```
clc; clear all

% Open source file
f1 = fopen('DATA/vbverslag.txt','r');
str = char(fread(f1)');
fclose(f1);

% Get headlines from source file
data = regexprep(str,'\s','xyz');
data = regexprep(data,'xyz', ' ');
data2 = regexp(data, 'HEADLINE:', 'split');
data3 = regexp(data2, '\s\s\s','once');

[x y] = size(data3);
dataset = cell(y-1,1);
for i=2:y
    dataset{i-1,1} = data2{i}(1:data3{i});
end

% Set all letters into lowercase. Remove punctuation, accents and double ...
    whitespace
dataset = lower(dataset);
blacklist = {'\.' ',' '?' '!' ':' ';' '-' '\' '/' '@' ')' '(' '_' '[' ']' ...
    char(34) char(39) char(145) char(146) char(147) char(148) char(8216) ...
    char(8217) char(8220) char(8221) char(8222)};
dataset = regexprep(dataset, blacklist, '');
blacklist_a = {char(224) char(225) char(226) char(227) char(228) char(229)};
dataset = regexprep(dataset,blacklist_a,'a');
blacklist_c = {char(230)};
dataset = regexprep(dataset,blacklist_c,'c');
blacklist_e = {char(232) char(233) char(234) char(235)};
```

```
dataset = regexprep(dataset,blacklist_e,'e');
blacklist_i = {char(236) char(237) char(238) char(239)};
dataset = regexprep(dataset,blacklist_i,'i');
blacklist_o = {char(242) char(243) char(244) char(245) char(246)};
dataset = regexprep(dataset,blacklist_o,'o');
blacklist_u = {char(249) char(250) char(251) char(252)};
dataset = regexprep(dataset,blacklist_u,'u');
blacklist_y = {char(253) char(255)};
dataset = regexprep(dataset,blacklist_y,'y');
dataset = regexprep(dataset,'\s\s',' ');
dataset = strtrim(dataset);

% Save formatted data in text file
f2 = fopen('DATA/verslag.txt','a');
if f2 < 0, error('Cannot open file'); end
fprintf(f2,'%s\n',dataset{:});
fclose(f2);
```

## B.2 application.m

This code consists of three cells. The first cell is to load the dataset into Matlab and transform the data into a matrix. In the second cell the query can be entered. The corresponding column in matrix $M$ is then removed and the class vector $N$ is made. At the third cell the `l1_logreg_train` method is applied. A detailed description of how the `l1_logreg_train` method works can be found on the website of the designers [6].

```
% Code to rewrite data to suitable format and analyse the data with the
% l1_logreg method (Steven Boyd)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Cell 1: Inlezen databestand                                       %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; close all; clc;

tic

% Read data file
f1 = fopen('DATA/vk_buitenland.txt','r');
if f1 < 0, error('Cannot open datafile'); end
data = textscan(f1,'%s','headerlines',0,'delimiter','\n','bufsize',10^6);
fclose(f1);

dataset = data{1};
totalexample = sprintf('%s ', dataset{:});
features = textscan(totalexample, '%s', 'delimiter', ' ');
features = features{1};
features = unique(features);
```

```
% Read top100 most common words and substract from dataset
f2 = fopen('STOP/stoplijstX.txt','r');
if f2 < 0, error('Cannot open common words file'); end
top = textscan(f2,'%s','headerlines',0,'delimiter',' ');
fclose(f2);
top = top{1};
top = unique(lower(top));
features = setdiff(features,top);

% Remove empty entries
features(cellfun(@(features) isempty(features),features)) = [];
dataset(cellfun(@(dataset) isempty(dataset),dataset)) = [];

m = length(dataset); % number of examples
n = length(features); % number of features

% Save features in text file
f3 = fopen('features.txt','wt');
if f3 < 0, error('Cannot open file'); end
fprintf(f3,'%s\n',features{:});
fclose(f3);

% Count the number of occurences, line by line
M = zeros(m,n);
for i = 1:m
    data = textscan(dataset{i}, '%s', 'Delimiter', ' ');
    example_regel = data{1};
    M(i,:) = cellfun(@(x) sum(strcmp(x, example_regel)), features);
end

toc

clear data dataset empty example_regel top totalexample;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Cell 2: Invoeren zoekwoord, verwijderen van zoekwoord uit dataset %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

M = sparse(M);

tic

% Search for a string and remove from dataset
str = 'obama';
loc_str = find(ismember(features,str) == 1);
if isempty(loc_str) == 1
    error('Word not found in dataset, try again')
end

% Make Class file
N2 = M(:,loc_str);
N = -1*ones(m,1);
```

```
N(N2>0) = 1;

features2 = features;
features2(loc_str) = [];
M2 = M;
M2(:,loc_str) = [];

toc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Cell 3: Toepassen methode                                         %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

tic

% Calling sequence for training
display('Training:')
mmwrite('x_data',M2);
mmwrite('b_data',N);

system('/home/ajacobs/Desktop/l1_logreg/l1_logreg_train -s -q x_data ...
    b_data 0.01 model');
model = mmread('model');

model2 = model(2:end);
loc_model = find(model2);
features3 = features2(loc_model);
waardes = nonzeros(model2);

% Save result in text file
f4 = fopen('RESULTAAT/res_features_china_vk.txt','wt');
if f4 < 0, error('Cannot open file'); end
fprintf(f4,'%s\n',features3{:});
fclose(f4);
f5 = fopen('RESULTAAT/res_waardes_china_vk.txt','wt');
if f5 < 0, error('Cannot open file'); end
fprintf(f5,'%s\n',waardes(:));
fclose(f5);

toc
```

# Bibliography

[1] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[2] J. Fox, *Applied Regression analysis and generalized linear models*, Sage, 2 ed., 2008.

[3] B. Gawalt, J. Jia, L. Miratrix, L. El Ghaoui, B. Yu, and S. Clavier, *Discovering word associations in news media via feature selection and sparse classification*, in Proc. 11th ACM SIGMM International Conference on Multimedia Information Retrieval, 2010.

[4] Instituut voor Nederlandse Lexicologie, *27 miljoen woorden krantencorpus 1995*. `http://www.inl.nl/`. Accessed July 2, 2011.

[5] K. Koh, S.-J. Kim, and S. Boyd, *An interior-point method for large-scale $l_1$-regularized logistic regression*. `http://www.stanford.edu/~boyd/papers/l1_logistic_reg.html`. Accessed February 14, 2011.

[6] ——, *l1_logreg: A large-scale solver for l1-regularized logistic regression problems*. `http://www.stanford.edu/~boyd/l1_logreg/`. Accessed March 5, 2011.

[7] M. Lieberman, *Logistic regression: Predicting the chances of coronary heart disease*. `http://www.slideshare.net/MultivariateSolutions/presentations`. Accessed August 17, 2011.

[8] NRC Handelsblad, de Volkskrant. `http://academic.lexisnexis.nl/tudelft/`. Accessed June 23, 2011.

[9] J. C. Pezullo and K. M. Sullivan, *Logistic regression*. `http://www.sph.emory.edu/~cdckms/Logistic/logistic.html`. Accessed August 17, 2011.