



Vrije Universiteit Brussel

Faculteit Wetenschappen  
Departement Computerwetenschappen

# Technieken binnen de Machine Learning voor Text Mining

Yannick Merckx

Vorbereiding op de bachelorproef

Rolnummer: 500294

Promotor: Yann-Michaël De Hauwere  
Begeleiders: Maarten Deville  
Peter Vrancx

Februari 2015



# Inhoud

<b>1</b>	<b>Introductie</b>	<b>2</b>
<b>2</b>	<b>Machine Learning</b>	<b>3</b>
2.1	Wat is Machine Learning . . . . .	3
2.2	Supervised Learning . . . . .	3
2.2.1	Regressie Probleem . . . . .	4
2.2.2	Classificatie Probleem . . . . .	7
2.3	Unsupervised Learning . . . . .	7
<b>3</b>	<b>Text Mining</b>	<b>8</b>
3.1	Document Pre-processing . . . . .	8
3.2	Methoden . . . . .	10
3.2.1	Vector Space Methode . . . . .	10
3.2.2	Probablistic methode . . . . .	12
3.3	LSA Experiment . . . . .	12
<b>4</b>	<b>Conclusie</b>	<b>13</b>

# Chapter 1

## Introductie

In deze voorbereiding gaat men technieken binnen de machine learning bespreken die men kan gebruiken voor text mining. Eerst gaat men een introductie geven over wat machine learning juist inhoudt, welke algemene technieken er worden gebruikt en waar men rekening mee moet houden bij deze technieken. Vervolgens gaat men specifiekere technieken bespreken, met de focus op text mining. Als laatste gaat men kijken hoe men deze technieken kan koppelen aan de eigelijke bachelorproef namelijk gevoelsanalyse op sociale media.

## Chapter 2

# Machine Learning

Machine learning is een welgekend begrip in de informatica wereld, maar wat het juist omvat, welke algemene technieken er bestaan en met welke factoren men moet rekening houden, wordt besproken in dit hoofdstuk.

### 2.1 Wat is Machine Learning

Over Machine Learning vindt men nergens een eenduidige definitie. Vele hebben geprobeerd om een eenduidige definitie te definiëren. Arthur Samuel(1959) definieerde machine learning als “Field of study that gives computers the ability to learn without being explicitly programmed”. Later stelde Tom Mitchell(1999) een well-posed learning problem als “A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ .” Als men machine learning wil omvatten, kan men het best omschrijven als een onderzoeksdomein dat zich bezighoudt met het onderzoeken en de ontwikkeling van zelflerende algoritmes. Hoofdzakelijk bestaat machine learning uit drie stappen namelijk data verzamelen, verwerken en analyseren.

Binnen machine learning kan men verschillende groepen van lerende algoritmes onderscheiden. Zo heeft men supervised learning, unsupervised learning, reinforcement learning en recommender systems. In deze voorbereiding gaat men zich enkel opleggen op supervised en unsupervised learning. Deze soorten algoritmen omvatten specifiekere technieken die zich lenen tot het gebruik bij text mining.

### 2.2 Supervised Learning

Wanneer men een algoritme wil trainen, heeft men informatie nodig om het algoritme te trainen. Dergelijke informatie noemt men de trainingset.

Laat men als voorbeeld een trainingset met positieve en negatieve artikelen nemen. Men weet welke artikelen positief en negatief zijn en ieder artikel bevat deze kennis aan de hand van een label. Het algoritme kan de informatie van de

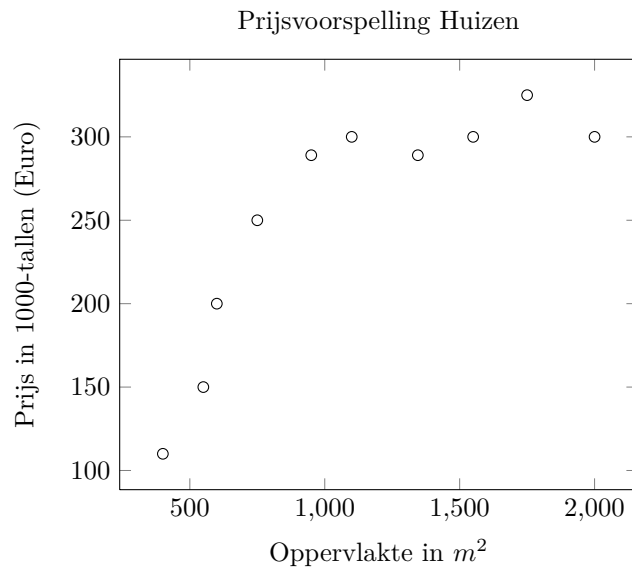
labels gebruiken om een zeker kennis te vergaren over artikels in het algemeen. Na het doorlopen van de trainingset, kan het algoritme de vergarde kennis gebruiken om een ongelabelde artikel te situeren als een positief of negatief artikel.

Deze techniek waarbij men een algoritme traint met een data waarvan men de antwoorden al weet noemt men supervised learning.

Het zelfstandig beslissingen maken over ongekende data is niet altijd even gemakkelijk en kan voor problemen zorgen. Bij supervised learning zijn er twee soorten problemen die kunnen optreden: een regressie probleem of een classification problem.

### 2.2.1 Regressie Probleem

Het doel dat men wil bereiken met supervised learning is dat het algoritme na een training antwoorden kan bezorgen over ongekende data. Bij het voorspellen van die antwoorden kan men te maken hebben met een regressie probleem. Dit probleem valt het best uit te leggen aan de hand van een voorbeeld. Neem nu dat men de prijs van een huis wilt voorspellen. Het algoritme traint zich met een trainigset en bekomt volgend resultaat als men zijn bevindingen zou plotten.



Stel nu dat men aan het algoritme de prijs van een huis met 1225 vierkante meter vraagt. Deze waarde zat niet in de dataset en moet dus voorspeld worden. Maar welke trend moet men volgen om de waarden te voorspellen. Men kan zowel kiezen voor een rechte of een 2de orde polynoom. Beiden zijn een mogelijkheid, maar geven een verschillend antwoord. De situatie, waarbij men een continue waarde moet bepalen en geen echte discrete afbakening bestaat, noemt men een regressie probleem.

Om dit probleem op te lossen, kan men van de techniek *lineaire regressie*

gebruik maken.

### Lineaire regressie

Lineaire regressie is een techniek waarbij het algoritme een hypothese probeert te vormen. De hypothese is een functie die opgesteld is aan de hand van de trainingsset en de gekende en ongekende outputwaarden zo goed mogelijk benaderd.

Als we terug kijken naar het voorbeeld van het huis. Kan het algoritme volgende hypothese opstellen.

$$H_{\theta}(x) = \theta_0 + \theta_1 x$$

Gegeven hypothese is een lineaire functie met als parameters  $\theta_0$  de nulconditie en  $\theta_1$  de richtingscoëfficiënt. Een hypothese met één functie noemt men ook wel een één dimensionale lineaire regressie.

Het opstellen van de hypothese introduceert op zijn beurt een **minimalisatie probleem**. Men moet de hypothese zo goed mogelijk opstellen, zodat de afwijking ten op zichte van de gekende resultaten minimaal is. Als de hypothese minimaal is, kan men er van uit gaan dat de afwijking op ongekende resultaten ook minimaal is.

Het minimalisatie probleem kan opgelost worden met een **kost functie** en **graduele afdaling**.

### Kost Functie en Graduele afdaling

Men herneemt het voorbeeld van de prijsvoorspelling van huizen. Men moest een zo precies mogelijke prijs voorspellen voor een oppervlakte van  $1225 \text{ m}^2$ . Om dit probleem op te lossen gaat het algoritme gaat voor zowel rechten als 2de orde polynomen de kostfuncties berekenen. Dit gebeurt door de prijzen van de gekende oppervlaktes te vergelijken met de prijzen van de hypothese. Door telkens het verschil in prijs voor een bepaalde oppervlakte te nemen, deze op te tellen en het gemiddelde te nemen, verkrijgt men de gemiddelde afwijking van de prijs van de hypothese ten op zichte van de echte prijs. Hierdoor krijgt men een beeld over hoe de prijzen van de hypothese zich verhouden tegenover de eigelijke prijzen. De kost functie voor dit voorbeeld is de functie met als functiewaarden de gemiddelde afwijkingen voor telkens een andere hypothese. Om dan een zo precies mogelijke voorspelling te kunnen doen voor de oppervlakte van  $1225 \text{ m}^2$  moet men er voor zorgen dat men een hypothese kiest waarbij de gemiddelde afwijking zo laag mogelijk is.

Algemeen kan men de kost functie definiëren als een functie die voor een bepaalde waarden van de parameters de gemiddelde afwijking van de hypothese ten opzichten van de resultaten gaat berekenen.

Volgende formule kan men opstellen voor de kost functie:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (H_{\theta}(x_i) - Y_i)^2$$

Deze kost functie noemt men ook wel de *squared error cost function* en wordt over het algemeen het meest gebruikt. Merk op dat men niet zomaar telkens de som van het verschil tussen het resultaat van de hypothese neemt en de eigelijke waarden. Het kwadraat van het verschil wordt genomen vanwege de negatieve verschillen die ook moeten worden opgenomen als afwijking. Verder vereenvoudigt men het rekenwerk door te delen door twee (De helft van de kleinste waarde, blijft de kleinste waarde).

Zoals eerder gezegd is het de bedoeling om de afwijking zo klein mogelijk te houden. Om het minimum van de kost functie te vinden, kan men de techniek *graduele afdeling* gebruiken. Omwille van verschillende redenen is graduele afdaling een van de meest gebruikte technieken binnen machine learning voor minimalisatie. Zo werkt de techniek voor een algemeen kost functie met  $n$  parameters  $J(\theta_0, \theta_1, \theta_2, \theta_3, \dots, \theta_n)$  en kan het altijd uitgevoerd worden aangezien de lineaire regressie kost functie altijd convex is.

De techniek start met een random start punt te nemen. Vervolgens gaat men stapsgewijs proberen te dalen tot je convergeert naar een lokaal minimum.

De preciese werking van het algoritme valt het best uit te leggen aan de hand van een voorbeeld. We nemen als voorbeeld onze eerder opgestelde hypothese met twee parameters  $\theta_0$  en  $\theta_1$ . Als men de kost functie  $J(\theta_0, \theta_1)$  berekent en deze weergeeft in een driedimensionale weergave, krijgt men onderstaande plot.

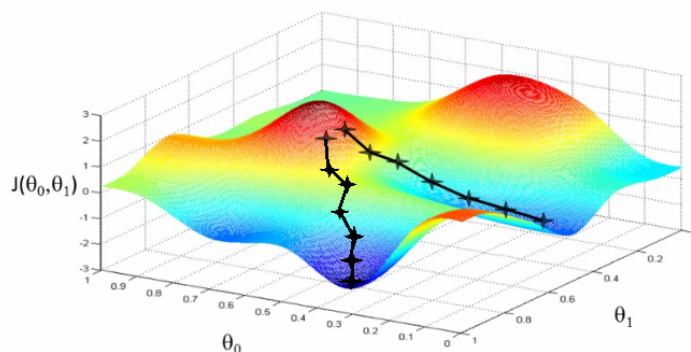


Figure 2.1: Driedimensionale weergave van de kostfunctie en zijn parameters

Het stapsgewijs dalen kan als volgende formeel neergeschreven worden:

$$\theta_j := \theta_j - \alpha \frac{d}{d\theta_j} J(\theta_0, \theta_1) \quad (\text{voor } j = 0 \text{ en } j = 1)$$

Alfa noemt men hier de learning rate. Dit is de grote van de stappen die men neemt bij het afdalen. De learning rate is een belangrijk element in het graduele afdalingsalgoritme. Als men deze te groot neemt kan men lokale minima overslagen en convergeert het algoritme niet. Als men alfa te klein neemt kan het algoritme heel lang duren.

Een belangrijk en subtiel detail bij de formule en het algoritme is het simultaan

updaten van de twee parameters (zowel  $\theta_0$  als  $\theta_1$ ). Als men dit niet doet, spreekt men niet van graduele afdaling.

Een bedenking die men moet maken bij graduele afdaling is het bestaan van meerdere lokale minima. Dit kan men echter eenvoudig oplossen door meerdere keren het algoritme uit te voeren met een ander startpunt.

Het gegeven voorbeeld noemt men specifiek ***batch graduele afdaling*** waarbij men telkens bij iedere stap de hele trainingsset vergelijkt. Er bestaan ook niet batch versies van graduele afdaling.

### 2.2.2 Classificatie Probleem

Een classificatie probleem is een ander soort van probleem dat zich voordoet bij supervised training. Een classificatie probleem doet zich voor wanneer men data moeten verdelen over verschillende discrete klassen en ieder element maar tot één klasse mag behoren. De classificatie kan gebaseerd zijn op één attribuut, maar ook op meerdere.

In het algemeen wordt het classificatie probleem het meest opgelost door ***logistische regressie***

## 2.3 Unsupervised Learning

Unsupervised learning is een techniek waarbij het algoritme zelfstandig moet leren hoe het juist moet en deze kennis gebruikt om later patronen en structuren in data te herkennen. De trainingsset bevat niet de antwoorden.

Als men het zou vergelijken met het voorbeeld van supervised learning, zou men bij unsupervised learning als voorbeeld de situatie kunnen nemen waarbij het algoritme een ongelabelde trainingsset van artikels krijgt en na het verwerken van deze artikels zelfstandig keuzes kan maken over welke artikels positief en negatief zijn.

Echter kan een algoritme de structuren en patronen herkennen, maar kan het niet de data concreet identificeren. Dit probleem kan men oplossen door gebruik te maken van cluster algoritmes. Concreet gaat een cluster algoritme de data groeperen of ***clusteren*** in groepen en zo de data concreet identificeren.



## Chapter 3

# Text Mining

Nu men een algemeen begrip heeft van wat machine learning juist is en welke algemene technieken het omvat, kan men overgaan naar text mining en zijn geschikte technieken. In dit hoofdstuk gaat men bespreken welke technieken men kan gebruiken voor text mining en wat deze juist inhouden. Als laatste gaat men de theorie toepassen op een voorbeeld en gaat men de resultaten van dit experiment bespreken.

Text mining of text data mining is een techniek waarbij men aan tekstanalyse doet om zo trends en patronen te kunnen vaststellen. Neem opnieuw als voorbeeld onze artikels. Met text mining wil men de artikels zodanig analyseren zodanig dat men kan uitmaken welk artikel positief en welk negatief is. Een probleem dat zich onmiddellijk bij text mining voordoet is het ontbreken van een één-op-én relatie van woorden en een concept. Woorden verwijzen zelfden eenduidig naar één concept. Zo het voorkomen van het woord "bank" in een tekst zowel verwijzen naar de financiële instelling als naar een doodgevone zitbank in het park. Dergelijke dubbele betekenis van woorden maakt het moeilijk om de woorden, met als gevolg ook de tekst, te mappen op een bepaald concept. Verder heeft men ook woorden in een tekst die weinig bijdragen tot de bepaling van het concept van de tekst bijvoorbeeld: ik,en,want... Deze woorden kan men uit de tekst filteren door een database aan te leggen met woorden die moeten men moet negeren. Deze techniek en nog soortgelijke alternatieven vereisen dat er al een voorverwerking plaatsvindt voordat men de dataset echt gaat analyseren op patronen en trends. Algemeen kan men zeggen als men de resultaten van de text mining wil optimaliseren, men aan *document pre-processing* moet doen.

### 3.1 Document Pre-processing

Document pre-processing is een optionele, maar zeker nuttig stap in het text mining proces. Document pre-processing bestaat eruit om je dataset al eens te verwerken, zodanig je extra informatie hebt, die je kan gebruiken bij de eigelijke analyse van de dataset. Zo kan je bijvoorbeeld alle stopwoorden verwijderen uit

de dataset. Wanneer men dan op deze gewijzigde dataset een analyse uitvoert, geeft men indirect de informatie mee dat stopwoorden er niet toe doen. Uiteraard is het verwijderen van stopwoorden één van de technieken. Er bestaan nog andere technieken die nuttig zijn als voorverwerking van een dataset. Zo kan men tekst en stucturen afleiden. Bijvoorbeeld het omzetten van Microsoft Word of Latex documenten naar XML maakt het parsen en analyseren van de documenten voor het algoritme veel gemakkelijker. Verder kan men ook **stemming** toepassen. Stemming is een techniek waarbij men tracht om de stam van het woord te achterhalen. Bijvoorbeeld uit het woord *katachtig* kan men het woord *kat* afleiden. De techniek kan gebaseerd zijn op een woordenboek bijvoorbeeld *Mmorph* is zo'n stemming woordenboek ontwikkeld door de Universiteit van Genève. Verder kan men de stemming ook baseren op een set van regels, bepaald door taalkundige. Het onderstaande voorbeeld illustreert een set van stemming regels voor het Frans:

$$\begin{aligned}
 (m > 0) \text{ aux} &\rightarrow al \\
 (m > 0) \text{ ouse} &\rightarrow ou \\
 (m > 0) \text{ eille} &\rightarrow eil \\
 (m > 0) \text{ nne} &\rightarrow n \\
 (m > 0) \text{ fs} &\rightarrow v
 \end{aligned}$$

Figure 3.1: Voorbeeld van stemming regels in het Frans

Tenslotte is **named entity recognition** (NER) ook een techniek die men kan gebruiken bij document pre-processing. Hierbij gaat men entiteiten proberen detecteren in de tekst en deze labelen. Neem bijvoorbeeld de zin *Yannick heeft zich ingeschreven de richting Computerwetenschappen aan de Vrije Universiteit Brussel in 2012*. Men kan met NER de entiteiten eruit halen, labelen en volgend resultaat verkrijgen: *[Yannick]<sub>persoon</sub> heeft zich ingeschreven de richting Computerwetenschappen aan de [Vrije Universiteit Brussel]<sub>organisatie</sub> in [2012]<sub>tijdsaanduiding</sub>*

Algemeen ziet men dat al deze technieken samen worden gecombineerd, wat alleen maar de uiteindelijke resultaten ten goede komt. Hoe deze gecombineerd kunnen wordt in het onderstaande voorbeeld geïllustreerd.



Figure 3.2: Combinatie van technieken bij document pre-processing

## 3.2 Methoden

Na de document pre-processing kan men beginnen aan de eigenlijk analyse van de dataset. Voor de text mining kan men 2 methodes gebruiken: de vector space methode en probabilistic methode.

### 3.2.1 Vector Space Methode

De vector space methode is een methode waarbij men in principe een document als een vector voorstelt waarbij ieder element overeenkomt met een woord en zijn frequentie in het document. Als men concreet een document voorstelt kan men zeggen dat document  $j$  voorgesteld wordt door  $\mathbf{d}_j$  met  $f_{ij}$  de frequentie van het woord  $w_i$ . Het aantal woorden stelt men voor door  $n_w$ , wat eveneens de dimensie is van de vector. Het document kan dus als volgt worden voorgesteld:

$$\mathbf{d}_j = \begin{bmatrix} f_{1j} \\ f_{2j} \\ \vdots \\ f_{n_w j} \end{bmatrix}$$

Een belangrijk inzicht bij het vector space methode is dat een document voorgesteld wordt als een groep van woorden. Er wordt geen rekening gehouden met de volgorde waarin de woorden in het document voorkomen. Vaak ziet men ook dat de vector vaak ijl is en vanwege de grote hoeveelheid aan woorden in een document heel groot. Als men nu niet één document maar meerdere documenten neemt en men zegt dat het aantal documenten gelijk is aan  $n_d$ . Dit resulteert in een matrix waarbij iedere kolom een document voorstelt.

$$D = \begin{matrix} & \text{Documenten} \\ \begin{matrix} f_{11} & f_{12} & \cdots & f_{1n_d} \\ f_{21} & f_{22} & \cdots & f_{2n_d} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n_w 1} & f_{n_w 2} & \cdots & f_{n_w n_d} \end{matrix} & \begin{matrix} \\ \\ \\ \end{matrix} & \text{Woorden} \end{matrix}$$

Deze matrix wordt ook een **terms-documents matrix (TDM)** genoemd. Wanneer men spreekt van een **documents-terms matrix (DTM)**, spreekt men een getransponeerde terms-documents matrix. Een rij van een DTM stelt dan een document voor. Maar hoe brengt deze voorstelling ons dicht bij het vinden van verbanden tussen de documenten? Wel men kan aan de hand van de euclidische afstand bepalen of documenten gelijkaardig zijn of niet. Stel men heeft twee documenten met een kleine euclidische afstand. Dit wil eigenlijk zeggen dat de vectorvoorstelling van de documenten gelijkaardig is. Wat wil zeggen dat de woordfrequenties ongeveer overeen komen en dus bijvoorbeeld de documenten over hetzelfde onderwerp gaan of eenzelfde mening uitdrukken. In de praktijk is gebleken dat documenten vergelijken op basis van woordfrequentie nog niet de gewenste resultaten opleverd. Vaak is het nog altijd moeilijk om verschillend

groepen tussen de documenten te onderscheiden. Daarom kan men nog extra verfijningen toepassen aan de hand van *term weighting* en *Latent Semantic Models (LSM)*.

### Term weighing

Als men even stil staat bij onze TDM met woordfrequenties, kan men zeggen dat niet elk woord evenveel doorweegt. Een woord dat in alle documenten voorkomt biedt geen of minder waardevolle informatie, dan een woord dat zelden voorkomt. En hierop baseerd term weighing zich. Het gaat een wegingsfactor introduceren. Ieder woord krijgt een gewicht toegewezen, dat weergeeft hoe belangrijk het woord is. Neem als voorbeeld een hoop recensies van de film "Pulp Fiction" en de woorden "Pulp" en "excellent". "Pulp" is een woord dat voorkomt in de titel van de film en komt ongetwijfeld in elke recensie voor. "Excellent" daarin tegen is een woord dat enkel maar voorkomt wanneer de recensist de film fantastisch vond, het zal niet in elk document voorkomen en is waardevolle informatie. Term weighting zal dus bij dit voorbeeld "excellent" een grotere gewicht toewijzen dan "Pulp". De quantiteit van dit gewicht wordt vaak de **inverse document frequency (idf)** genoemd en wordt bepaald aan de hand van volgende formule:

$$w_i : idf_i = -\log_2[P(w_i)]$$

met  $P(w_i)$  dat priori probability dat word  $w_i$  voorkomt in het document In woorden geeft de inverse document frequency het algemeen belang van het woord  $w_i$  weer. Men kan dit benaderen door het logaritme te nemen van het aantal documenten waar  $w_i$  in voorkomt en het totaal aantal documenten. Een andere nuttige quantiteit is de **term frequency**  $tf_{ij}$ . Deze geeft het belang weer van het woord  $w_i$  binnen in het document  $d_j$  en wordt als volgt genoteerd:

$$tf_{ij} = \frac{f_{ij}}{\sum_{i=1}^{n_w} f_{ij}}$$

Met deze twee quantiteiten kan men een nieuwe begrip introduceren: de textbf-idf score. Wat overeenkomt met het product van tf en idf.

$$tf-idf \text{ score} = tf.idf_{ij} = idf_i.tf_{ij}$$

De tf-idf matrix bekomt men dan door alle woordfrequenties van het terms-document matrix te vervangen door de tf-idf score. Deze matrix wordt bijvoorbeeld vaak gebruikt om de gelijkenissen tussen twee documenten te bepalen op basis van cosinusgelijkenis

### Latent Semantic Models

Als tweede verfijning van het vector space model, heeft men latent semantic models (LSM). Met LSM probeert men een notie te krijgen van de semantische informatie en maar bepaald het semantisch verband tussen woorden. Bijvoorbeeld als men zoekt naar documenten met het woord "economie", men ook documenten

met "financiën" zou terugkrijgen. Voor LSM zijn twee woorden semantisch gerelateerd als ze gebruikt worden in dezelfde context. Met het concrete voorbeeld kunnen we zeggen dat er een semantisch verband is tussen 2 woorden als ze vaak voorkomen in hetzelfde documenten.

Merk op dat bij Latent Semantic Models het wederom belangrijk is dat ieder woord, naar één concept verwijst. Analystisch wordt LSM toegepast door **Singular Value Decomposition (SVD)** toe te passen op de terms-document matrix. SVD is een concept uit de lineaire algebra en gaat gegeven een rank  $m$  de matrix met rank  $n$  zo goed mogelijk benaderen, resulterend in een matrix met rank  $m < n$ . Concreet betekent dit dat men een document, voorgesteld als een vector met  $n$  woorden, kan transformeren naar een vector met  $m$  getallen. De getallen van de vector die het document voorstellen, noemt men ook wel **features**. Doordat men de dimensionaliteit van de vectoren kan beperken door semantisch gelijkaardige woorden bijeen te voegen. Laat dit toe om een soort van context groepen te creëren en zo een zeker inzicht te krijgen in de dataset.

### 3.2.2 Probablistic methode

Een probablistic methode gaat statistiek gebruiken om zo inzicht te krijgen over de data. Ieder user profile  $u_k$  wordt voorgesteld door een statistisch model. Een document kan relevant ( $R = 1$ ) of niet relevant ( $R = 0$ ) zijn voor een user. Die relevantie kan men bepalen op basis van een gestorterd vector space model. Op basis van die ordening in het vector space model gaat men de kans berekenen dat het document relevant is voor  $u_k$ .

Men kan formeel de kans dat een gegeven document  $d = x$  relevant is voor user profile  $u_k$  formeel neerschrijven als

$$P(R = 1 | d = x, u_k)$$

Hoe grotere deze waarde, hoe grotere de kans dat document  $x$  relevant is.

Een voorbeeld van een probalistic methode is de naïve bayens classifier. Deze kijkt naar de kans dat men een document beschouwd dat relevant is voor user  $u_k$ . Dit wordt als volgt neergeschreven:

$$P(d_n = x_n | R = 1, u_k)$$

Dit is zeer gemakkelijk te berekenen door te kijken naar de verhouding tussen de documenten met woord  $w_n$  en de relevante documenten. Analoog kan men ook de kans dat men een document beschouwd dat niet relevant is voor user  $u_k$  bepalen.

Merk op, een model waarbij men twee staten heeft, relevant en niet relevant, noemt men een binary independence probabilistic retrieval model. Dit model gaat er vanuit dat alle woordvoorkomens onafhankelijk zijn.

## 3.3 LSA Experiment

## Chapter 4

# Conclusie

Nu men weet wat machinelearning juist omvat, welke technieken er zijn en hoe men specifiek text mining moet aanpakken, kan men dit koppelen aan de bachelorproef. De bachelorproef omvat het onderzoeken en toepassen van een gevoelsanalyse op sociale media. Sociale media is in wezen één grote set aan data. Men kan besluiten dat Machine learning de oplossing biedt. Het bevat alle middelen om de gevoelsanalyse op sociale media toe te passen. Om succesvol een gevoelsanalyse toe te passen moet men volgende stappen ondernemen. Ten eerst moet men de data verzamelen. Aangezien men te maken heeft met een dataset waarvan men geen informatie heeft, moet men technieken gebruiken van unsupervised learning. Vervolgens moet met de data preprocesen met het LSA algorithm. Het experiment heeft uitgewezen dat dit een zeer krachtig algorithm is, dat men meer inzicht geeft in de data. Ten slotte moet men de data verwerken met een cluster algorithm. De sequentie van verzamelen, verwerken en analyseren is een zeer belangrijk gegeven bij onze gevoelsanalyse. CONCLUSIE KAN NOG BETER EN MISSCHIEN UITGEBREIDER, MAAR DIT ZIJN DE OUTLINES VAN DE CONCLUSIE