Listening to the Flock

Towards opinion mining through data-parallel, semi-supervised learning on social graphs.

June 2014

Submitted to the Department of Mathematics and Computer Science in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science

Christophe Van Gysel
chris@stophr.be

Prof. dr. Bart Goethals
bart.goethals@uantwerpen.be

Samenvatting

Sociale media is niet langer weg te denken uit de hedendaagse maatschappij. Ze heeft voorgoed de manier veranderd waarop personen onderling communiceren en interageren. Haar snelle verspreiding en alomtegenwoordige aard motiveert het gebruik voor sociale, politieke en commerciële doeleinden. Gebruik van deze diensten neemt snel toe. Deze groei brengt eveneens een stijging in de hoeveelheid gestructureerde en ongestructureerde informatie teweeg die door deze diensten verzameld wordt.

De grote hoeveelheid data beschikbaar voor onderzoek heeft toepassingen in vakgebieden zoals pattern mining, recommendation systems en anderen. De vraag of computers over een cognitieve capaciteit kunnen beschikken gelijkaardig aan die van de mens staat centraal in de Informatica. Alan Turing stelde de vraag *"Can machines think?"* reeds in 1950. Tot op heden blijft deze vraag onbeantwoord.

Digitale computers zijn in staat om informatie sneller en op grotere schaal te verwerken dan mensen. Tijdens de Amerikaanse verkiezingen in 2012 speelde de analyse van persoonsinformatie een belangrijke rol voor het modelleren van kiezersgedrag. Dankzij deze modellen waren partijen in staat om zwevende kiezers te identificeren. Dit werk richt zich op een vergelijkbaar doel waarbij we publieke informatie van het sociale netwerk Twitter verzamelden over gebruikers in Vlaanderen. Vervolgens gebruiken we deze informatie om de verschillende politieke invloeden in hun sociale netwerken te achterhalen. Het doel van het werk ligt op het verwerken van gegevens op een grote schaal en hoge snelheid zonder menselijke tussenkomst. Hoewel een persoon in sommige gevallen betere verbanden kan ontdekken in informatie, is het moeilijk om dit te verwezenlijken op gelijkaardige schaal.

In december 2013 hebben we gegevens van Twitter verzameld van ongeveer 12 000 gebruikers. Van elk van deze gebruikers verkregen we hun volgers en tweehonderd meest recentste tweets via openbare Twitter informatiebronnen. Op basis van deze gegevens induceerden we een graafstructuur gebaseerd op de interacties tussen gebruikers, met name keken we naar gebruikers die een andere gebruiker volgen en gebruikers die tweets van een andere gebruiker retweeten.

Vervolgens identificeerden we de Twitter accounts van Vlaamse politieke partijen en naarmate hun locatie in de social graph wisten we de overige gebruikers een kansdistributie over deze partijen toe te wijzen. Meer bepaald keken we naar de kans dat een random walk over de social graph terecht kwam bij een politieke partij, telkens beginnende bij een willekeurige gebruiker. Voor alle gebruikers opgenomen in het onderzoek verkregen we dus een verdeling over de voornaamste Vlaamse politieke partijen. Het is belangrijk om op te merken dat de enige voorkennis van het systeem, specifiek tot het Vlaamse politieke landschap, de Twitteraccounts waren van acht politieke partijen. Alle informatie en voorspellingen over Vlaamse kiezers werd dus afgeleid van deze minimale domeinkennis.

Om onze resultaten te beoordelen verkregen we lijsten van Twitter accounts van politiek geëngageerden. Deze lijsten werden gepubliceerd door de Vlaamse politieke partijen op Twitter. We veronderstellen dat gebruikers die deel uitmaken van deze lijsten gelinkt zijn met de partij die ze beschikbaar stelde. In totaal verkregen we een validatie set van 700 gebruikers. Na uitvoering van de random walk vergeleken we voor elke gebruiker in de validatie set de sterkste partij in hun individuele verdelingen met de partij waarmee ze gelinkt werden. Onze voorspelling is correct voor gemiddeld 85% van de politiek geëngageerden. Bijkomend merkten we op dat als een voorspelling verkeerd was, ze nog steeds relatief dichtbij viel in het politiek spectrum tegenover de gelinkte partij. Bijvoorbeeld werden gebruikers geassocieerd met een extreemlinkse partij voorspeld te behoren tot een meer gematigde linkse strekking.

Onze resultaten kunnen echter niet gebruikt worden als een voorspelling van de verkiezingsuitslag. De Twitterpopulatie is nu eenmaal geen representatieve voorstelling van de Vlaamse kiezers. Bijkomend merken we ook op dat de politieke instelling van politiek geëngageerden nu eenmaal eenvoudiger te achterhalen is dan die van overige gebruikers. Vanwege het stemgeheim is het moeilijk om een validatie set te bekomen waarvan eveneens niet-politiek geëngageerde gebruikers deel uitmaken.

In mei 2014, een week voor de Vlaamse verkiezingen, publiceerde de Universiteit Antwerpen in samenwerking met de onderzoeksgroep ADReM een persbericht over het onderzoek verricht in dit werk. Gelijktijdig met het persbericht werd ook een website (twitterbrengtraad.be) gelanceerd waarop individuele gebruikers hun resultaten konden opvragen. Het bericht werd verspreid door de meeste grote Vlaamse nieuwswebsites, verscheen in een aantal Vlaamse kranten (Metro, Het Laatste Nieuws en De Standaard) en werd besproken op de Vlaamse publieke omroep (Radio 1).

Abstract

In this thesis we consider the application of graph classification in order to determine the political climate of the social circles of Twitter users. To accomplish this we crawl Twitter, a microblogging service, and retrieve connections and interactions between users. We then induce a weighted, directed graph structure from this data. Markov random walks are used in order to obtain a probability distribution over political parties, effectively modelling the distance between users and political parties. We implement the Adsorption algorithm by Baluja et al. [2008] in the MapReduce paradigm using Apache Crunch. Further we investigate the importance of certain features on the microblogging platform with respect to our problem domain. We find that retweets on Twitter are a valuable indicator of like-mindedness. Reciprocal connections, however, do not seem to exhibit this property. We achieve F_1 scores of 0.85 and higher over a validation set of politically engaged Twitter users.

We conclude that this method lends itself to valuable knowledge extraction. We believe that performance will only increase as models become more advanced and more data becomes available.

Acknowledgments

Foremost, I would like to express my gratitude to my supervisor, Prof. Bart Goethals, for guiding and assisting me while working on this thesis. Secondly I would like to thank the Mathematics and Computer Science department at the University of Antwerp for the excellent education I received there. Time and time again they challenged and motivated me to keep performing at a high level. I would not have been the computer scientist I am today without their guidance.

During my time at the University of Antwerp I was an active member of the Mathematics, Physics and Computer Science organization for students. The experiences I have gained through involvement in the board of the organization played an enormous role towards my self-development, organizational skills and leadership qualities. I would like to thank all members of the organizational board over the years I was actively involved with the organization.

Over the past year I have gained invaluable work experiences that had a big impact on the work I performed in this thesis. In particular I would like to thank Gus Katsiapis, Samuel Ieong and Sam Slee at Google for the time I spent there. At Facebook, my gratitude goes towards Alec Muffet, Chad Parry and Abhishek Doshi. All these people guided me during my internships and made me the software engineer and researcher I am today.

I would also like to thank my family for the support they provided me throughout my entire life. Without the support of my parents, Marc and Catherine, and my brother, Cedric, I would most likely not have started my undergraduate studies at the University of Antwerp. Let alone would I have been able to obtain my masters degree as the result of this thesis.

Last but not least, I would like to thank Joachim Ganseman and Benjamin Allardet-Servent for all their suggestions while helping me proof-read this thesis.

Table of Contents

Chapter 1

Introduction

Social media is changing how humans connect and interact with each other [Junco et al., 2011]. Its fast phase of expansion and omnipresent nature motivates its use in various social, political and commercial settings [Howard, Hinsliff, Young]. Usage of these web services has been increasing steadily [Kiss, 2014]. With its growing adaptation come large amounts of structured and unstructured data. The analysis of this data can lead to interesting discoveries and applications.

In order to preserve user privacy only some amount of this data is publicly available. Online social networks offer users fine-grained control over what information can be seen by whom. The availability of data also depends heavily on the vision behind the services. For example, by default all messages on Twitter can be seen by anyone. Other services, such as Facebook, offer a more private model of communication where user relations require reciprocal confirmation and messages are often limited to the first degree of separation.

Despite these different models of communication, the amount of data available for research is unprecedented and has uses in fields such as pattern mining [Nohuddin et al., 2011], recommendation systems [Resnick and Varian, 1997] and others. The notion of machines demonstrating a cognitive ability equal to that of humans has been an area of research (*Can machines think?*) that goes back to the origin of digital computing [Turing, 1950]. This remains an open question in the field of Computer Science.

Machines are able to process information at a much higher rate than humans, which has been shown useful in applications such as machine translation. During the 2012 United States elections the analysis of personal information played an important role [Duhigg] by effectively modelling voter behavior. Using data gathered from different sources they were able to identify voters that were still in doubt. This thesis focuses on a similar cause where we will gather public information on Twitter about Flemish users. We will use this information to determine the political climate of their social circles.

More precisely, we present a simple graph model focusing on user connections and interactions. The model is built using connection and interaction data retrieved from Twitter. Previous research [Boyd et al., 2010] has shown that retweets are a valuable feature. From this model we then infer the relative distance between each user and eight Flemish political parties using Markov chains and random walks. The distance between each user and the respective political parties is influenced by how close their neighbors are situated to these parties. This relation is solved recursively for each user individually until all parties are reached. The applied method is somewhat similar to the PageRank algorithm by Page et al. [1998]. After presenting the model we retrieve a distribution over political parties for these users. This distribution indicates the political climate of their Twitter circles. Our focus lies on the social circles of individual users and does not represent a representative sample of the active population in Flanders.

We validate our predictive performance using a validation set that exists of 700 Twitter accounts of Flemish politically active users. These accounts were retrieved from lists published by the political parties on Twitter themselves. When comparing the predicted political climate of these accounts with the ground truth labeling we observe good performance values for recall and precision.

In May of 2014 the University of Antwerp published a press release in collaboration with the Advanced Database Research and Modelling (ADReM) research group on the work performed in this thesis. At the same time a website was unveiled where individual users were able to look at their own results. The news was picked up by the most notable Flemish news sites and was discussed on a public broadcasting radio talk show. More information regarding the media attention received by this work can be found in Appendix A.

The next chapter discusses the mathematical basis used in this thesis. In Chapter 3 we explore the Twitter online social network and its various aspects. Chapter 4 presents an algorithm by Baluja et al. [2008], some implementation-specific consequences and how our model performance will be assessed. In Chapter 5 we show our experiments, results and performance measurement over a validation set of 700 politically active users. Chapter 6 talks about possible extensions to this thesis and possible uses of its results. Finally, we summarize and conclude this thesis in Chapter 7.

Chapter 2

Labeling graphs

Graphs are versatile and important data structures due to their ability to model relations between objects [Diestel, 2000]. Graph theory, and in particular the study of graph algorithms, is a significant field within computer science. It encompasses a variety of interesting theoretical problems, some of which have real-world applications.

Modelling the usage of communication systems is one problem where graph theory triumphs. Over recent years the available information about the usage of these systems has increased thanks to advancements in technology and increasing transparency [Saramäki and Onnela, 2007]. As mentioned in the introduction, the rise of social media has uncovered new ways for people to interact, relate and communicate with each other. These novel communication mediums are interesting for social [Young], economical [Kaplan and Haenlein, 2010] and political [Howard, Hinsliff] objectives.

The growing popularity of social media has raised questions regarding its influence on society and interpersonal relations. The ability to analyze sentiments or mine opinions from text through the application of machine learning methods has been an active area of study [Pang and Lee, 2008]. In this thesis a similar analysis of sentiment is considered, based on the structure of social graphs rather than the contents of messages. We focus on the propagation of labels in these graphs, similar to how ideas spread in social circles.

This chapter gives an overview of the mathematical background and theory used in this thesis. These constructs, definitions and approaches were originally defined by Szummer and Jaakkola [2002], Zhu et al. [2003], Zhu [2005] and Azran [2007] and are mentioned here for completeness and convenience.

1 Semi-supervised learning

When collecting data it requires little effort to gather large amounts of unlabeled data. One of the recurring problems in machine learning is performing classification of previously unseen instances, given a set of example instances from which a machine is expected to learn. Obtaining a training set is considered difficult as labeling instances is an expensive process. It often requires human annotators with experience in the problem domain. Semi-supervised learning addresses this problem by using large amounts of unlabeled data together with a smaller set of ground truth data in order to build better classifiers [Zhu, 2005].

There is no free lunch. Compared to traditional supervised learning, a smaller set of labeled instances is required. Because of the lack of labeled data, semi-supervised learning methods make strong assumptions about the model. Consequently, users of semi-supervised learning methods should spend a reasonable amount of time and effort designing models [Zhu, 2005]. Various semi-supervised learning methods exist and there is no optimal or generic solution. Therefore it is recommended to use a method whose assumptions fit the problem structure.

In 2005 Zhu published a survey regarding the literature on semi-supervised learning. As most of it is outside of the scope of this thesis, interested readers are encouraged to read up on the matter in Zhu [2005]. More information on semi-supervised learning can be found in the book by Chapelle et al. [2010]. More information on the broad subject of machine learning, data mining and information retrieval can be found in Mohri et al. [2012], Hastie et al. [2008] and Manning et al. [2008].

2 Representing data as a labeled graph

Given a set of n objects, we construct a graph $G(V, E, W)$ where V is the set of n nodes representing the objects, E is the set of edges and W is the edge weight matrix. In many cases these connections are inherent to the problem domain and may already be well-defined.

2.1 Labels

Classification of a graph implies some labeling of either nodes or edges. This thesis focuses on the labeling of nodes $v_i \in V$ [Bhagat et al., 2011]. A data set might already contain labels for some v_i, or a limited set of examples can be labeled by an oracle (e.g. a human annotator). Let V_l be the set of l initially labeled nodes and V_u the set of $n - l$ unlabeled nodes [Azran, 2007].

Let $\mathcal{Y}$ be the set of m classes with $m \geq 2$. Every node v_i should eventually carry a distribution $P(Y = c|i)$ over $\mathcal{Y}$ ($c \in \mathcal{Y}$). Nodes in V_u have no distribution at first. Denote $\{y_1, ..., y_l\}$ as the set of initial label distributions assigned to the nodes in V_l. These distributions form the rows of the $l \times m$ stochastic matrix Y_l such that row i contains the distribution y_i [Bhagat et al., 2011]. Define Y as the $n \times m$ matrix where the first l rows are equal to the rows of matrix Y_l and the remaining rows consist of only zero.

The definition for a *label connected graph* is given by Azran [2007]. In order to provide a classification for nodes in V_u at least one labeled node should be reachable from any initially unlabeled node.

Definition 1 (Label connected graph). *A weighted, directed graph $G(V, E, W)$ is said to be label connected if for every node $v_i \in V$ there exists at least one path of strictly positive edges between v_i and a node in V_l, the set of initially labeled nodes.*

After classification of $v_i \in V$ it is often desirable to select a single label from every label distribution. To label a node v_i a common choice [Bhagat et al., 2011] is to choose the label c_i that maximizes the node's label distribution:

$$c_i = argmax_c P(Y = c|i) \tag{1}$$

Having a distribution over labels potentially allows for multi-label classification, where a single instance can be assigned multiple labels. In this case one can select the top-k labels that maximize the distribution. Another approach is to select only frequent labels, denoted by $\{c \in \mathcal{Y} | P(Y = c|i) \geq \frac{1}{m}\}$ for each $v_i \in V$.

2.2 Nodes, edges and weights

For the purpose of this thesis a weight matrix W is required where W_{ij} denotes the directed edge from node i to node j (a weight of zero indicates a non-existing edge). We divide the weight matrix W into four blocks by splitting after the l-th row and column [Zhu et al., 2003].

$$W = \begin{bmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{bmatrix} \tag{2}$$

The decomposition of W implies an ordering of the rows in the matrix, as well as the size of each sub-matrix. For example, W_{lu} is of size $l \times u$ and it includes the weights of all edges from labeled to unlabeled nodes [Azran, 2007].

We can induce a graph if no network structure is defined over the objects. Given a data set $X = \{x_1, ..., x_n\}$ in some metric space S, a distance function $d : S \times S \to \mathbb{R}$ and a weighting function $w(\cdot; \sigma)$ where σ is a vector of parameters [Zhu et al., 2003]. The weight matrix can be defined as

$$W_{ij} = w(d(x_i, x_j); \sigma) \tag{3}$$

for every pair of objects $(x_i, x_j) \in X^2$. W_{ij} is symmetric due to the symmetry property of distance metrics. Solely the definition of weight matrix W is sufficient for performing classification by the methods described in later chapters.

We now briefly discuss the example of using the Euclidean norm as distance function. Afterwards we look at the possibility of only considering the k-nearest neighbors of every node.

Euclidean space When S is some Euclidean space of dimension d (e.g. $\mathbb{R}^d$), use of the Euclidean norm $d(x_i, x_j) = ||x_i - x_j||_2$ as a distance metric comes naturally. A possible weight matrix W is

$$W_{ij} = exp\left(-\sum_{k=1}^{d} \frac{(x_{ik} - x_{jk})^2}{\sigma_k^2}\right) \tag{4}$$

where d is the size of the vector space, x_{ik} is the k-th component of vector x_i and the vector σ represents length scale parameters, usually taken as the variance [Zhu et al., 2003]. From Equation 4 and the identity property of metrics it is know that $\forall i, j : 0 < W_{ij} \leq 1$ and $W_{ii} = 1$ for all i (due to the normalizing behavior of the σ vector). G is a complete graph as every pair of distinct nodes is connected by a unique edge, consequently G is label connected if at least one node in G carries an initial label distribution.

K-nearest neighbors Exponentially weighting edges such as in Equation 4 is convenient to highlight the importance of close-by nodes. This approach may result in a dense graph representation with a high number of insignificant connections.

An alternative is to only consider the k-nearest neighbors of every node v_i where closeness is to be determined by a distance function (as in Section 2.2.2), resulting in a sparse, non-symmetric weight matrix [Bhagat et al., 2011].

2.3 A note on social graphs

Inducing a graph from n data points using the method described in Section 2.2.2 has a time complexity of $O(n^2)$ as the pairwise distance has to be calculated for every pair of data points (x_i, x_j). In the context of large-scale data mining any algorithm that runs in super-linear time complexity is undesirable and should be avoided if possible.

The *social graph* is the global mapping of everyone and how they are related [Brad Fitzpatrick, 2007]. The analysis of social networks is based on an assumption of the importance of relationships among interacting units [Wasserman and Faust, 1994]. The rise in popularity of online social networks (such as Facebook and Twitter) provides the opportunity to study the characteristics of online social networks at large scale [Mislove et al., 2007].

The underlying graphs of these social networking sites often depict actors as nodes who are linked together via multiple interaction contexts or affiliations [Kossinets, 2003] and can be either directed or undirected. The edges between nodes are generally assigned weights in such a way that a greater weight indicates a stronger relationship.

In order to measure the strength of relations in these graphs some concepts from the fields of network analysis and sociology are useful to take into account.

- In the broad context of link-based classification, Lu and Getoor [2003] suggest identifying types of hypertext regularities that are applicable to the problem domain. Yang et al. [2002] describe the concepts of *encyclopedic regularity*, which claims that linked objects typically have the same class, and *co-citation regularity*, where objects citing the same object tend to have the same class.
- *Homophily* is the principle that a contact between similar people occur at a higher rate than among dissimilar people [McPherson et al., 2001]. It implies that distance in terms of social characteristics translates into network distance. Lazarsfeld and Merton [1954] make the distinction between status homophily and value homophily. The former refers to individuals with similar social status being more likely to associate with each other, whereas the latter applies to individuals who think in similar ways.

3 Graph labeling problem

Given a graph $G(V, E, W)$ as defined in Section 2.2, a subset of labeled nodes $V_l \subset V$ (Section 2.2.1) and $V_u = V - V_l$ the set of unlabeled nodes. Labels $\tilde{Y}$ are inferred on all nodes V in the graph.

We wish to find a mapping between nodes V and the set of labels $\mathcal{Y}$.

$$\pi : V \rightarrow \mathcal{Y} \tag{5}$$

The task at hand is to find a mapping π such that some objective function is optimized given a parameter vector σ where σ represents the configuration of the classification model (see Section 2.2.2 for an example).

4 Markov chains and random walks

Markov chain A *Markov chain* is a collection of random variables $\{X_t\}$ having the property that, given the present, the future is conditionally independent of the past [Weisstein, a].

$$P(X_t = j|X_0 = i_0, X_1 = i_1, ..., X_{t-1} = i_{t-1}) = P(X_t = j|X_{t-1} = i_{t-1}) \tag{6}$$

Stochastic matrix A *stochastic matrix* is the transition matrix for a finite Markov chain. Elements of the matrix must be real numbers in the closed interval $[0, 1]$ [Weisstein, d]. Consider a Markov chain $X_0, X_1, ...$ with discrete state space E of size n. The Markov chain is specified by the $n \times n$ transition matrix $P = (p_{ij})_{i,j \in E}$ (see Equation 7), where row $p_i = (p_{ij})_{j \in E}$ is a transposed probability vector for every state $i \in E$ [Asmussen, 1987].

State transition For a Markov chain $\{X_t\}$ calculate the probability $\mathbb{P}(X_0 = i_0, X_1 = i_1, ..., X_t = i_t)$ that the chain transitions over a sequence of states $i_0, ..., i_t$, ending up in state i_t and starting from the initial distribution μ [Asmussen, 1987].

$$\mathbb{P}(X_0 = i_0, X_1 = i_1, ..., X_t = i_t) = \mu_{i_0} \cdot p_{i_o i_1} \cdot p_{i_1 i_2} \cdots p_{i_{t-1} i_t}$$

At time t the chain is restarted with the new initial value X_t. The post-t chain $X_t, X_{t+1}, ...$ evolves as the Markov chain itself, started at X_t but otherwise independent of the past [Asmussen, 1987].

Matrix power Consider a square $n \times n$ matrix A. The power A^n of a matrix A for a nonnegative integer is defined as the matrix product of n copies of A [Weisstein, b].

$$A^n = \underbrace{A \cdots A}_{\text{n}}$$

For example, from Equation 6 it can be seen that the sequence $X_0, X_m, X_{2m}, ...$ is a Markov chain with transition matrix P^m [Asmussen, 1987].

Random walks on graphs A random walk on a graph is a random process consisting of a sequence of discrete steps of fixed length [Weisstein, c]. Given a directed graph $G(V, E, W)$ as described in Section 2.2 the transition matrix can be computed as

$$P = D^{-1} W \tag{7}$$

where D is a diagonal matrix $D = diag(d_i)$ such that $d_i = \sum\limits_j W_{ij}$ [Bhagat et al., 2011, Szummer and Jaakkola, 2002, Azran, 2007] and W is the edge weight matrix Equation 3.

The Markov chain $X = (X_0, X_1, X_2, ...)$ with state space V and transition matrix P is a random walk on the graph G. This chain governs a particle moving along the nodes of G. If the particle is at $v \in V$ at a given time, then the particle will be at a neighbor of v at the next point in time. The neighbor is chosen randomly, in proportion to the weight prescribed by the transition matrix [Siegrist, 2001].

If an irreducible and aperiodic Markov chain is described by a single time-independent matrix P then it has been shown that after a large enough number of steps the steady-state probabilities $P^{\infty} = \lim_{t \to \infty} P^t$ of the Markov chain are reached [Szummer and Jaakkola, 2002, Feller, 1968].

Take the initially labeled nodes V_l as absorbing states, that is a state i such that $p_{ii} = 1$, then rewrite Equation 2 as follows:

$$W = \begin{bmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ W_{ul} & W_{uu} \end{bmatrix}$$

where symbols $\boldsymbol{I}$ and $\mathbf{0}$ reference the identity and null matrices respectively. Note that the choice of unit weight in W_{ll} is arbitrary. As the transition matrix is the normalized weight matrix:

$$P = \begin{bmatrix} P_{ll} & P_{lu} \\ P_{ul} & P_{uu} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ P_{ul} & P_{uu} \end{bmatrix}$$

If G is label connected (Definition 1) then P^{∞} is given by

$$P^{\infty} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ P_{ul}^{\infty} & \mathbf{0} \end{bmatrix}$$

due to the absorbing nature of the label distributions of nodes in V_l. Every node $v \in V$ ends up with a distribution over the initially labeled nodes V_l [Azran, 2007]. The matrix equation for graph labeling using random walks [Bhagat et al., 2011] can then be written as

$$\tilde{Y} = P^{\infty} Y \tag{8}$$

where Y is the $n \times m$ matrix from Section 2.2.1, and

$$\tilde{Y_u} = P_{ul}^{\infty} Y_l. \tag{9}$$

In other words, every node $v \in V_u$ receives a distribution over $\mathcal{Y}$ that is a linear combination of its distribution over the initially labeled nodes.

5 Alternative approaches

In this thesis we apply random walks on graphs to obtain a labeling over nodes. This method works well for the purpose of determining the political climate of the social circles of Twitter users. First of all, the number of political parties is limited and most parties are represented on the social network. Furthermore Twitter is actively being used for political campaigning [Hinsliff]. Additionally voting is compulsory in Flanders, we therefore assume that every Flemish citizen has some political preference.

In this section we briefly discuss data mining alternatives for graph labeling. These methods come from unsupervised and supervised learning contexts.

5.1 Iterative Classification Method

Bhagat et al. [2011] note that if two objects are related it is possible that inferring something from one object can aid inferences about the other. They present an iterative classification approach that exploits this characteristic in relational data [Neville and Jensen, 2000].

They state that there are multiple ways to approach classification in a relational context. For example, you can simply ignore all relational data and build classifiers using local node features only. Alternatively one can incorporate features introduced by the relations when building models. Possible features include the number of neighbors of a specific class. In the *Iterative Classification method* models are rebuilt and feature vectors updated each iteration. On every iteration all nodes are re-classified using the new model, until convergence is achieved.

5.2 Clustering

Clustering algorithms can be used to find partitionings of data. In our application the number of clusters can be taken as the number of political parties. Unfortunately due to its unsupervised nature most clustering algorithms do not allow specification of the cluster centers upfront. The *k-means* or *k-medians* algorithms could be used by specifying the political parties as initial cluster centers. It does however not guarantee that the cluster centers are retained. Use of the k-means method in this way comes close to a semi-supervised learning approach, as the learning is not strictly unsupervised any longer.

5.3 Semi-supervised learning

In this thesis we only apply a single method from the semi-supervised learning field. There are various other methods in the field of semi-supervised learning that could be applied [Zhu, 2005, Chapelle et al., 2010]. Even when only considering methods similar to the one applied here, there are still many possible variants.

Chapter 3

Modelling the Twitterverse

Twitter is a microblogging service that launched in 2006. Users of the service send and receive text messages limited to 140 characters, also known as *tweets*.

Twitter's growing popularity has led to it being used in a number of ways, such as for educational purposes [Junco et al., 2011], in political campaigning [Hinsliff], in case of emergencies [Young] and more. Around 2011 the use of the microblogging service increased during the so-called *Arab Spring* as a coordination platform for protests [Howard].

1 On Twitter and microblogging

Microblogging is a form of blogging that lets users publish short text-based updates. Kaplan and Haenlein [2011] attribute the success of microblogging to its notion of ambient awareness, push-push-pull communication and virtual exhibitionism.

Twitter allows for a unique type of communication. If a user finds the tweets of another user valuable, they can decide to become a *follower* of that person. All followers of a particular user receive all that user's tweets on a personalized homepage. *Following* implies a one-directional relationship from one user to another and indicates that a user is interested in another user's messages.

Tweets are public and undirected by default. This means that any user can view or interact with them. The large number and transient nature of these messages causes many tweets to go by unnoticed. If a message is deemed interesting by the viewer they may decide to give the message an extra push by retweeting it to their own followers, effectively propagating the message through the network as word-of-mouth.

Tweets can also be directed to another user while remaining publicly visible. This often leads to open discussions on the social network. Users also have the ability to adjust privacy controls and lock down their accounts such that only reciprocal followers can interact with them. In addition, users have the ability to send private messages to each other.

This leads to the question of whether a relationship brought forward by one-directional or reciprocal following on Twitter contains a notion of trust. Interpersonal ties in social network often carry this characteristic, such as in mobile telecommunications networks [Saramäki and Onnela, 2007]. Some suggest that reciprocal following on Twitter has little meaning in terms of interpersonal tie strength [Scholle, Braun] over one-directional relationships. For example, people would follow each other as a token of appreciation and to increase their collective follower count. This implies the absence of trust in these relationships [Dougherty].

Granovetter [1973] remarks the importance of weak interpersonal ties in a social networks with respect to the propagation of information through the network. The strength of a tie is best-regarded as a linear combination of the amount of time, emotional intensity, intimacy and the reciprocal services which characterize the tie [Granovetter, 1973].

In this chapter we focus on retrieving a sizable data sample from Twitter. Subsequently we discuss the induction of a graph model for use in classification. Retrieving a representative sample from the network is not an easy task due to imposed throughput restrictions. It is important to determine a sampling strategy that limits the scope of the sample to relevant users only. The platform represents a social graph by users establishing one-directional connections between each other. While these connections expose relations between users, they do not indicate their strength. In order to weight these connections other events on the network are considered.

2 Data gathering

Twitter provides a programming interface (API) that allows for the querying of information and interacting with the service [Twitter, a]. For example, it is possible to retrieve all tweets and followers of a specific user through this API. Unfortunately the API is rate-limited. This prevents us from making many requests within a short period of time. The limits imposed by the service depend on the type of request. Therefore sampling is necessary in order to retrieve a subgraph of the social network.

In February 2014, Twitter announced a pilot project through which they gave research institutions access to their full data repository [Krikorian, 2014]. Unfortunately the program was not available during development of this thesis.

Implications of rate-limiting Let us consider the time it takes to retrieve the followers of n users. At the time of writing the programming interface allows for the retrieval of 5 000 followers per request. Subsequently the service limits users to 60 such requests per hour, or one request per minute. Considering that the average user has 208 followers [Roberts] then we are able to process one user per minute. Note however that some users have follower numbers that go in the hundreds of thousands. Due to their well-connected nature this type of users are frequently encountered when crawling the graph.

The problem of sampling large graphs has been documented by Leskovec and Faloutsos [2006]. Leskovec and Faloutsos discuss how to derive a representative graph sample using different sampling methods, while taking into account the sample size, and how to extrapolate measurements of the sample. Ye et al. [2010] cover the applied use case of crawling online social networks.

Sampling method Existing work suggests traditional graph strategies (e.g. breadth-first search), choosing nodes based on their degree and sampling nodes or edges randomly. In particular Ye et al. [2010] make the distinction between $V_{observed}$, the set of nodes the crawler knows exists, and $V_{selected}$, the set of nodes sampled, where $V_{selected}$ is a subset of $V_{observed}$.

For the purposes of node classification it is desirable to retrieve a label connected sample (Definition 1). To attain this property in the sample graph, we initialize the set of labeled nodes V_l as seed nodes during crawling. Denote F_i as the set of followers of user i. The union of all followers of the seed nodes may be too large, given limited computing resources or timing constraints. Merely mutual followers with a certain support are considered:

$$supp(U, F_l) = \frac{|\{F \in F_l | U \in F\}|}{|F_l|} \tag{10}$$

where U is a candidate user and $F_l = \{F_i\}$ for all labeled nodes $v_i \in V_l$.

3 Graph induction

In Section 2.2.2 we discussed how to induce a graph structure for objects x_i given some distance function. For a large amount of objects and streaming interaction data we prefer an alternative approach.

The tie strengths are taken as a linear combination of various interactions [Granovetter, 1973]. Consider a sample of size n and l distinct features. A feature is an interaction between users on the social network. For every user i in our sample

$$W_i^\top = \begin{bmatrix} I_{1i,1} & I_{2i,1} & \dots & I_{li,1} \\ I_{1i,2} & I_{2i,2} & \dots & I_{li,2} \\ \vdots & \vdots & \ddots & \vdots \\ I_{1i,n} & I_{2i,n} & \dots & I_{li,n} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_l \end{bmatrix} \tag{11}$$

where I_{kij} is the count of directed interaction I_k from user i to user j for all users $v_i, v_j \in V$. w_k is the independent weight assigned to interaction I_k. These weights $\{w_1, ..., w_l\}$ constitute the parameters of the model.

The following sections discuss some of the most frequent interactions on Twitter and their relevance when modelling tie strength in social graphs.

3.1 Profiles

Every user of the service has a profile with information such as their location, a profile picture, a description and more. While this information does not generally describe ties between users, these ties can be inferred by application of named entity recognition. The co-citation regularity principle states that users referring the same object are more likely to be similar [Yang et al., 2002].

3.2 Followers and friends

The connections made by users following each other lay down the basic structure of the social graph. As suggested by the concepts homophily and encyclopedic regularity (see Section 2.2.3), similar users are more likely to connect and interact with one another [McPherson et al., 2001].

Reciprocal connections Saramäki and Onnela [2007] analyze the structure of mobile communication networks. They consider a reciprocal call of long duration between two users as a signature of some relationship. Evidence [Braun, Dougherty, Scholle] suggests that this does not apply on Twitter. During experimentation reciprocal connections will be considered separately in order to empirically measure the influence of these connections on the tie strength between users on Twitter.

3.3 Tweets

Tweets are the basic atomic building blocks of Twitter [Twitter, c]. In their purest form these messages are unstructured text and allow for the application of natural language processing methods.

Twitter encourages the use of metadata tags in tweets. These tags introduce structured elements in unstructured messages. These elements provide additional context and allow for the easy extraction of connections between entities.

Entities Referenced entities in tweets, such as resolved URLs, media, hashtags and user mentions offer additional context without having to parse text to extract that information [Twitter, b]. This suggests the application of sentiment analysis in order to determine the attitude of the speaker with respect to these entities.

For example, a user mention can be seen as an additional directed connection from the author to the mentioned user. The connection weight could then be inferred from the sentiment of the message. The principle of co-citation regularity can also be considered. These cited objects can be the URLs, media objects and hashtags embedded in tweets.

"What other people think" has always played an important part in the information-gathering behavior of humans [Pang and Lee, 2008]. Sentiment analysis is the task of identifying positive and negative opinions, emotions and evaluations [Wilson et al., 2005].

As mentioned in Section 3.1, Twitter is used in a broad number of ways by users from all over the world. It lends itself perfectly for the task of sentiment analysis, given its huge amount of subjective content. The results of Popescul et al. [2002] demonstrate that the combination of text and link features often improves performance.

However, it has been noted that the domain of microblogging is different from that which handles longer-text variants [Kouloumpis et al., 2011, Bermingham and Smeaton, 2010]. Models have been trained from Twitter data using supervised and unsupervised machine learning methods. Supervised methods used either a training set of hand-annotated tweets [Bermingham and Smeaton, 2010], or a training set retrieved from considering emoticons in tweets as labels [Pak and Paroubek, 2010]. Bermingham and Smeaton [2010] encountered poor performance when using *SentiWordNet*[1] for unsupervised learning of a sentiment analysis model on Twitter data.

Retweets Boyd et al. [2010] compare retweeting with email forwarding on a structural level. Users propagate other users' messages through the network. They state that the act contributes to a conversational ecology in which conversations are composed of a public interplay of voices.

They give an overview of possible reasons why people retweet, such as the validation of others' thoughts or to amplify tweets to new audiences. While those reasons are interesting for this thesis it has also been noted [Boyd et al., 2010] that some users retweet purely for the purpose of self-gain. The latter hope to gain followers or reciprocal retweets of other participants. Their work concludes that retweeting can be both a productive communicative tool and a selfish act by attention seekers.

[1] http://sentiwordnet.isti.cnr.it/

Chapter 4

Data-parallel learning

The linear algebra approach using Markov chains in Section 2.4 works for small graphs. However, social networks often consist of hundreds of thousands of nodes. The mathematical algorithm given in Section 2.4 is computationally expensive as matrix multiplication runs in cubic time complexity for dense matrices. Additionally, the denseness of these matrices depends greatly on the applied model.

MapReduce [Dean and Ghemawat, 2004] is a programming model for processing and generating large data sets. It significantly eases the task of developing data-parallel applications. These applications usually exist of a pipeline of MapReduce tasks of which the programming and management can become overwhelming. Flume [Craig Chambers, 2010] provides an API that makes it easy to develop, test and run efficient data-parallel pipelines.

Engineers at Google invented MapReduce and Flume. Although their implementation remains internal to the company, the open-source world has since then developed Apache Hadoop [The Apache Software Foundation, 2005] and Apache Crunch [Foundation, 2013]. These systems are an implementation of the papers published by Google and have become increasingly popular frameworks for data processing.

The problem formulation of random walks can be translated to an iterative approach which lends itself to be expressed in the MapReduce paradigm.

1 Iterative formulation of random walks

In the random walk formulation we consider the probability that a walk starting at an arbitrary node ends at the labeled nodes. It expects that the weight W_{ij} of a directed edge $v_i \rightarrow v_j$ in the adjacency matrix is proportional with the probability that a random walk follows that edge. However, in the iterative formulation the label is propagated through the network starting at the labeled nodes. The iterative version thus requires us to invert the edges such that a directed edge $v_j \rightarrow v_i$ means that v_j influences v_i.

The Adsorption algorithm presented by Baluja et al. [2008] is used in this thesis. During every iteration each node takes the weighted average of the labels of its neighbors. In addition they introduce shadow nodes which offer the possibility to express uncertainty about the provided labeled data. We denote L_v as the label distribution of node $v \in V$ during algorithm execution. Initially L_v is set for the nodes in V_l only.

Injection transition probabilities The random walk formulation in Section 2.4 considers a directed graph $G = (V, E, W)$, a set of labels $\mathcal{Y}$ and the set of labeled nodes $V_l \subset V$ that have been assigned an initial label distribution over $\mathcal{Y}$ (see Section 2.2.1). Assume a shadow node $\tilde{v}\ \forall\ v \in V_l$ such that $V' = V \cup \{\tilde{v}\}$, $E' = E \cup \{v \to \tilde{v}\}$ and $W_{v,\tilde{v}} = \delta_{\text{transition}}$. Furthermore the label distribution L_v is relocated from v to $\tilde{v}$ for every $v \in V_l$. Let $\tilde{V}$ denote the set of shadow nodes $\{\tilde{v} \mid v \in V_L\}$. Only nodes in $\tilde{V}$ are initially assigned a label distribution.

The value of $\delta_{\text{transition}}$ is the probability that a random walk at $v \in V_l$ transfers to the absorbing state $\tilde{v}$ [Baluja et al., 2008]. Intuitively this value can be seen as a confidence measure for the initial labeling and can be interesting when there are more seed nodes than class labels.

Adsorption via averaging For use in the iterative approach all edges of graph G need to be inverted. This means we have to transpose weight matrix W. The semantics of the transition probability $\delta_{\text{transition}}$ changes. It is likely that node $v \in V_l$ has multiple incoming edges and thus the choice of $W_{\tilde{v}v}$ should be chosen relative to the total incoming weight at node v.

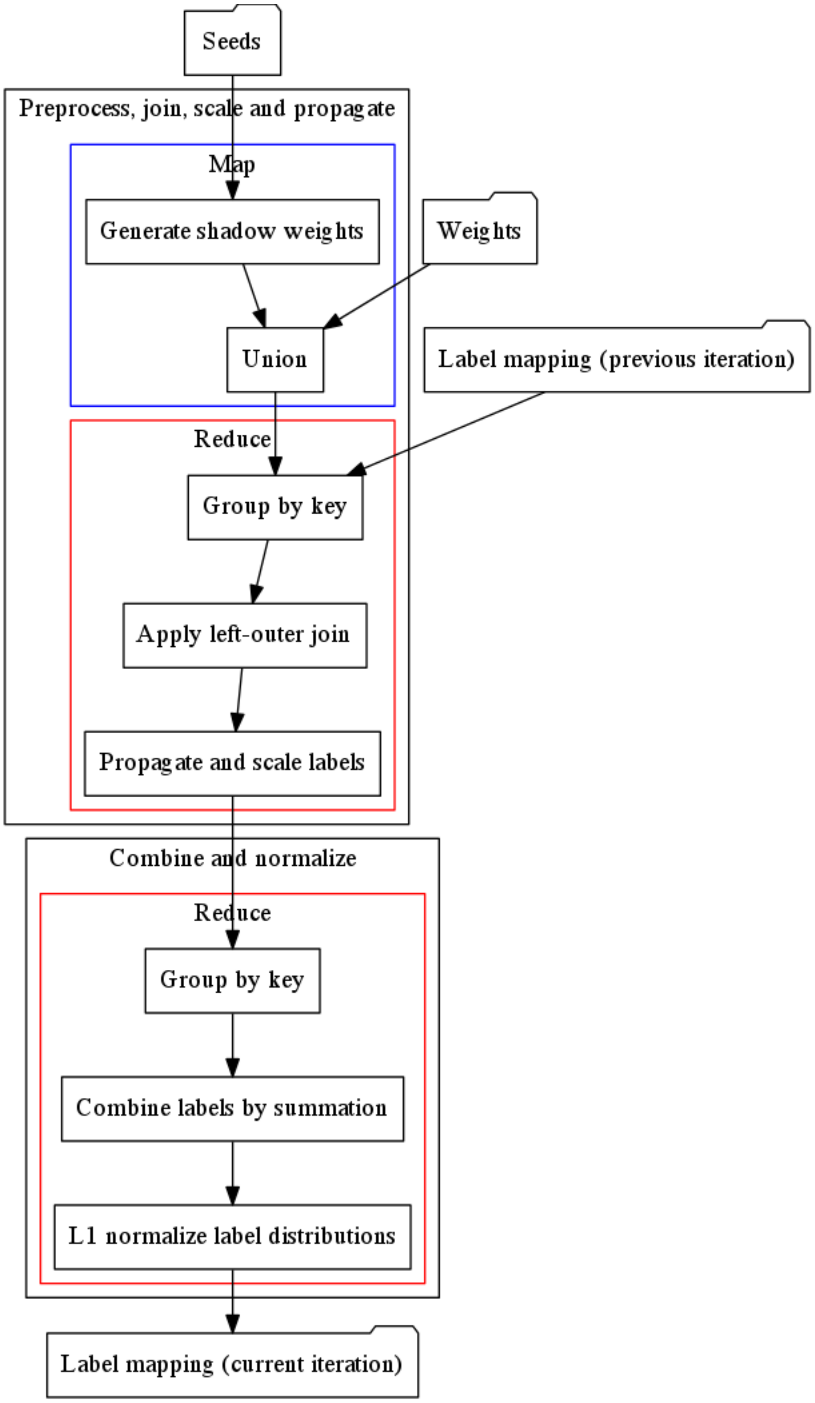
In the common case where $\delta_{\text{transition}} = 1$ one possibility is to set $W_{\tilde{v}v} = +\infty$ in the inverted graph G'_{inv}.

Algorithm 1: Adsorption algorithm [Baluja et al., 2008]. An iterative version of the random walk approach.

```
Input: $G'_{inv} = (V \cup \tilde{V}, E'_{inv}, W^\intercal), L$
repeat
    for $v \in V \cup \tilde{V}$ do
        $L_v \leftarrow \sum_u W^\intercal_{uv} L_u$
    end for
    Normalize $L_v$ to have unit $L_1$ norm
until convergence
Output: Distributions $L_v | v \in V$
```

1.1 Implementation

We implemented the algorithm from the previous section in the Apache Crunch framework. Crunch provides a programming interface similar to that in the work by Craig Chambers [2010]. This method provides a generic implementation that can be executed on various parallel computing platforms, such as Apache Hadoop and Apache Spark [The Apache Software Foundation, 2013].



Seeds
Preprocess, join, scale and propagate
Map
Generate shadow weights
Weights
Union
Label mapping (previous iteration)
Reduce
Group by key
Apply left-outer join
Propagate and scale labels
Combine and normalize
Reduce
Group by key
Combine labels by summation
L1 normalize label distributions
Label mapping (current iteration)

Fig. 1: Execution plan of a single iteration of the Adsorption algorithm on Apache Hadoop

Figure 1 shows a simplified representation of the execution plan generated for a single iteration of Algorithm 1. The execution plan shows how an iteration is split up in two MapReduce jobs when used with Apache Hadoop. Specific MapReduce-related details were left out of the execution plan.

Floating point arithmetic limitations The approximate nature of the finite representation of real numbers in floating point arithmetic leads to various limitations. As can be seen on line 3 of the algorithm, a large number of floating point numbers is aggregated by summing. To avoid large relative errors due to this operation these floating point numbers could be summed in ascending order according to their absolute value or make use of compensated summation algorithms.

Other caveats include catastrophic cancellation when calculating convergence (see Section 4.1.2) or loss of precision when multiplying small values (e.g. probabilities) during induction of the graph. The accuracy of the latter can be improved by first converting the values to log-space.

1.2 Convergence

Algorithm 1 runs until convergence of the label distribution. The label distribution converges to steady-state probabilities if the graph is label connected (see Definition 1). Any random walk over the graph will eventually encounter an absorbing state and end there. It has been shown that irreducible, aperiodic Markov chains have steady-state probability distributions [Feller, 1968] and thus converge as $t \to +\infty$.

In practice it is still good to verify its convergence. One way to do this is by making sure whether any node v_i has not changed label c_i (see Equation 1) between iterations. A practical approach is to count the number of nodes whose label distribution differ significantly between iterations. For instance, convergence of the graph labeling can be measured by counting nodes v_j that violate inequality

$$\left\| P(Y|j)^{(i)} - P(Y|j)^{(i-1)} \right\|_2 \leq \delta$$

where $P(Y|j)^{(i)}$ and $P(Y|j)^{(i-1)}$ are the label distributions for node v_j in iteration i and iteration $i-1$ respectively, and δ is some threshold value.

Non-converging behavior Due to its iterative nature the algorithm can get stuck in an oscillating state, where two heavily co-dependent nodes switch label distributions every iteration. This behavior, combined with the limitations of floating point numbers, often prohibits the algorithm from converging.

Self-loops can solve this problem in most cases. The effectiveness of this approach depends on the weight of the self-loops relative to the total weight of the incoming edges of a node.

2 Performance evaluation

Performance of graph classification is assessed using some well-known techniques from information retrieval. Results are aggregated in a data-parallel context and afterwards analyzed on a single machine.

2.1 Label distribution

The normalized sum over the individual label distributions $P(Y|i)$ for all nodes v_i results in the marginal distribution over labels $P(Y)$ (i.e. label distribution). We also consider the distribution of the classes that maximize the label distributions c_i (i.e. class distribution; see Equation 1). Comparison of these distributions can lead to interesting observations.

One interesting phenomenon is that in particular cases there is a remarkable difference between the aforementioned distributions. In this particular case the distribution of classes c_i has a single dominating class, while the probabilities of $P(Y)$ are more evenly-distributed. This behavior can be explained by neutral nodes adapting the majority vote propagated by the most influential class in the graph.

2.2 Confusion matrix

For every classification we consult its confusion matrix to evaluate model performance. A confusion matrix is a square $m \times m$ matrix where the rows show the number of instances in the actual class and the columns contain the number of instances in the predicted class. The matrix diagonal contains the number of instances for every class that were classified correctly.

In a multi-label problem there might be some overlap between classes. Even though an incorrect classification is undesirable, it might give insight in the problem domain and eventually lead to a better model.

A musical example Consider the problem of graph classification where a given graph exists out of songs and one wishes to classify these by their genre. It is generally more desirable to classify a metal song as rock, as opposed to classical music.

2.3 Recall and precision

Confusion matrices are a good way to asses some classification of objects. It is common to aggregate these instance counts in measures that assess performance over all classes.

For multi-class classification the concepts of recall and precision are generalized. For a single class c_i the number of true positives can be found on the diagonal of the confusion matrix (i.e. at position (i, i)). The instances that have been predicted to belong to c_i but in reality belong to another class are called the false positive instances (i.e. those on position $(i, j)\ \forall\ j,\ j \neq i$). The false negatives are those that actually belong to c_i but have been assigned a different predicted class (i.e. those on position $(j, i)\ \forall\ j,\ j \neq i$).

For every class c_i precision and recall [Manning et al., 2008] are defined as follows:

$$\text{Precision}_i = \frac{tp_i}{tp_i + fp_i} \tag{12}$$

$$\text{Recall}_i = \frac{tp_i}{tp_i + fn_i} \tag{13}$$

where tp_i, fp_i and fn_i denote the number of true positive, false positive and false negative instances of class c_i respectively.

Aggregate measures for multi-class classification can be obtained by combining these results through an average per-class agreement (macro-averaging, or M) or from sums of per-instance decisions (micro-averaging, or μ) [Manning et al., 2008]. The former considers all classes equal, while the latter considers classes that have more validation instances more important.

$$\text{Precision}_M = \frac{\sum_{i=1}^{n} \frac{tp_i}{tp_i+fp_i}}{n} \qquad \text{Precision}_\mu = \frac{\sum_{i=1}^{n} tp_i}{\sum_{i=1}^{n} tp_i + fp_i} \tag{14}$$

$$\text{Recall}_M = \frac{\sum_{i=1}^{n} \frac{tp_i}{tp_i+fn_i}}{n} \qquad \text{Recall}_\mu = \frac{\sum_{i=1}^{n} tp_i}{\sum_{i=1}^{n} tp_i + fn_i} \tag{15}$$

Recall and precision can be combined using the harmonic mean, also known as the F_1 score.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{16}$$

2.4 Confidence

It is possible that some nodes are classified with more certainty than others. We define a confidence of classification c_i in order to express this certainty.

Zhu et al. learn the weight matrix W by minimizing average label entropy [Zhu et al., 2003]. Entropy is a measure of uncertainty in a random variable [Ihara, 1993]. Let X be a discrete random variable with probability distribution $P(X)$, the entropy of random variable X is defined as

$$H(X) = -\sum_i P(x_i) \cdot \log_b P(x_i) \tag{17}$$

where $\{x_1, ..., x_n\}$ are the possible values of X and b is the base of the logarithm used. Strictly speaking Equation 17 is undefined if $P(x_i) = 0$ for some x_i. In that case the value of the corresponding summand $0 \cdot \log_b 0$ is taken to be 0.

The least predictable probability distribution, or the probability distribution with the most uncertainty, is the uniform distribution. This implies an upper bound on the entropy of discrete random variables. We rescale $H(X)$ to the interval $[0, 1]$ such that value close to 1 indicates low uncertainty.

For a distribution of labels $P(Y|i)$ over $\mathcal{Y}$ (as seen in Section 2.2.1) confidence is defined as

$$Conf(Y|i) = 1.0 + \frac{\sum_{j=1}^{m} P(Y = c_j|i) \cdot \log_b P(Y = c_j|i)}{log_b(m)} \tag{18}$$

where b is the base of the logarithm used (usually $b = 2$).

A common technique is to introduce different thresholds and re-evaluate performance using only those predictions that exceed the aforementioned threshold. More precisely the relevance measures are recomputed for the set

$$V_{conf} = \{v_i \in V | Conf(Y|i) \geq \delta_{conf}\} \tag{19}$$

which contains all nodes with a classification confidence higher or equal to the imposed threshold δ_{conf}. The set $V - V_{conf}$ is the set of non-selected nodes whose classification confidence is lower than δ_{conf}.

Chapter 5

Determining the political climate of the Flock

We performed experiments using a real-world sample collected from Twitter. The focus of the experiments lies on determining the political climate of the social circles of Twitter users situated in Flanders, a region in Belgium. Flanders was chosen as target area as it is the home of the University of Antwerp. In addition, it is relatively self-contained with a population of around 6 million people[2] and has a diverse multi-party political system.

1 Overview of the Flemish political spectrum

While Flanders is still a part of Belgium and the federal elections influence the country as a whole, residents can only vote on parties that are active in their region. Other regions in Belgium are Wallonia and the Brussels-Capital Region. While some parties overlap in regions, many of them operate in just one. Moreover, there are parties that are similar in political viewpoint but operate in different regions. The organization of the Belgian governments is complex, to say the least. Every region has its own government and so does every language community. These governments overlap in area, but each of them has its own responsibilities. The exact composition of governments in Belgium is out of the scope of this thesis, for more information see Jacobs, Deschouwer [2010].

For the purpose of this thesis eight parties active in the Flemish region were considered (the name between brackets is their English translation).

- Christen-Democratisch en Vlaams/CD&V (Christen Democratic and Flemish)
- Groen (Green)
- Libertair, Direct, Democratisch/LDD (Libertarian, Direct, Democratic)
- Nieuw-Vlaamse Alliantie/N-VA (New Flemish Alliance)
- Open Vlaamse Liberalen en Democraten/Open Vld (Open Flemish Liberals and Democrats)
- Partij van de Arbeid van België/PVDA (Workers' Party of Belgium)[3]

[2] See http://statbel.fgov.be/nl/statistieken/cijfers/bevolking/structuur/woonplaats/

[3] PVDA is a minor party in Flanders. They were included because of their communist, left-winged view that overlaps with socialist parties.

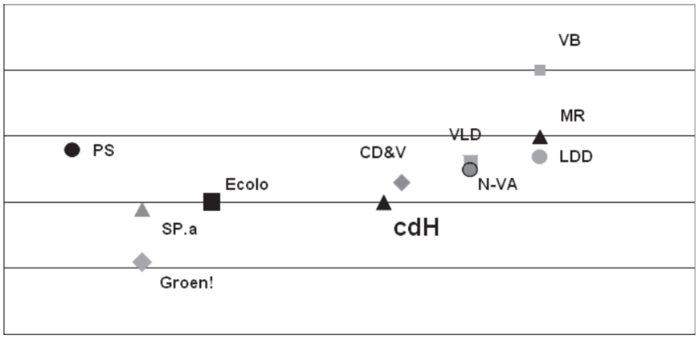


Fig. 2: The Belgian political spectrum according to Deschouwer [2010]. The horizontal axis represents the left-right positions, while the vertical axis depicts progressive-conservative mentality.

- Socialistische Partij Anders/sp.a (Different Socialist Party)
- Vlaams Belang/VB (Flemish interest)

Deschouwer [2010] gives his vision on the Belgian political spectrum. Although the political positions of the different parties may change over time and can depend on the point of view of the observer, the spectrum given in Deschouwer [2010] is used as a frame of reference in this thesis (see Figure 2).

Political party	**Twitter user**
CD&V	@cdenv (15753618)
Groen	@groen (8200752)
LDD[4]	@JMDedecker (166555933)
N-VA	@de_NVA (24688336)
Open Vld	@openvld (21994861)
PVDA	@pvdabelgie (82068939)
sp.a	@sp_a (23423414)
VB	@vlbelang (22473403)

Table 1: Overview of the Twitter accounts (with unique user identifier between parentheses) used for the political parties.

1.1 Social presence on Twitter

We used a representative Twitter account for each of the eight parties (Table 1). It should be noted that the usage of Twitter by these parties does not necessarily correlate with their popularity. Some parties encourage the use of social media, while others keep a distance. The use of the results of this study as a popularity poll between parties may seem attractive, however, it is not a good representation of the population of Flanders and should not be used to extrapolate. The goal of this study is not to predict the 2014 elections, but rather to model the Flemish political climate on Twitter from publicly available data.

Validation To validate and measure performance a validation set of users is essential. These were extracted from lists published by Twitter accounts of the political parties themselves. Validation sets for parties that did not publish such lists were retrieved from third party newspapers. The final validation set consists of 734 Twitter accounts (see Table 2).

While a validation set of decent size was gathered it is important to note that these users, given that they are Flemish politically engaged users, most likely export strong features related to political parties in Flanders. Therefore determining the social climate of their social circles on Twitter should render fairly good results.

[4] The Twitter account of the chairman was used as the official party account (@ldd_nationaal; 135161040) has been inactive since June 2012

[5] Validation list provided by the Twitter account of a large Belgian newspaper (@Tweetstraat16; 143755138).

Political party	**Validation list**	**Size**
CD&V	Volg CD&V (83417343)	69
Groen	twittersfeer (4140216)	125
LDD[5]	lijst dedecker (12865557)	12
N-VA	N-VA (97441623)	82
Open Vld	Open Vld (41055982)	177
PVDA	PVDA twitterverse (71150646)	21
sp.a	sp.a (436888)	274
VB	Vlaams Belang (12667247)	20

Table 2: Overview of validation lists (with unique list identifier between parentheses) consisting out of 734 Twitter accounts of Flemish politically active users.

2 Experimental set-up

Over the period from December 2013 to January 2014[6] we sampled 12 254 Twitter users that followed at least two of the aforementioned political parties. For each of these users their 200 most recent tweets were retrieved, resulting in a total of 1 249 091 tweets.

Information about the set of 12 254 selected users, denoted as $V_{selected}$ (see Section 3.2) of graph $G_{selected}$, was retrieved. This information included their user name and profile descriptions, in addition to a list of their followers. Other users, contained within this list but not part of $V_{selected}$ (including those that follow a user in $V_{selected}$), make up the set V_{others}. We then consider $V_{observed} = V_{selected} \bigcup V_{others}$ of graph $G_{observed}$. Graph $G_{observed}$ does not contain any additional connection data than that gathered for the purpose of graph $G_{selected}$ and a large number of users in $V_{observed}$ (around 10 million) may therefore be influenced by a small set of people. For example, if thousands of users of $V_{observed}$ follow a single user $v \in V_{selected}$ then each of these users will copy the label distribution of user v. Therefore we only consider $G_{selected}$ when evaluating performance.

For each experiment the graph $G_{observed}$ was induced (Section 3.3) and the algorithm (Chapter 4) ran until convergence. During graph induction only a subset of the social interactions described in Section 3.3 were considered:

- when user i follows user j, add weight $w_{follows}$ to W_{ij}, conversely add $w_{followed}$ to W_{ji},
- when user i reweets user j, add weight $w_{retweets}$ to W_{ij}, conversely add $w_{retweeted}$ to W_{ji},
- when user i and user j follow each other, add $w_{reciprocal}$ to W_{ij} and W_{ji}.

[6] Sampling took this long due to rate-limiting constraints enforced by Twitter.

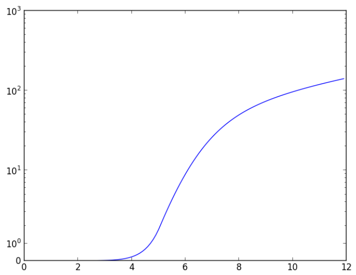


Fig. 3: Logarithmic scale plot of the weighting function ($\alpha = 6.0$, $\beta = 2.0$) in the $[0.0, 12.0]$ interval.

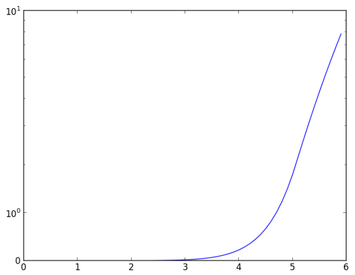


Fig. 4: Logarithmic scale plot of the weighting function ($\alpha = 6.0$, $\beta = 2.0$) in the $[0.0, 6.0]$ interval.

Note that the previous weightings are applicable in the random walk formulation. For use in the Adsorption algorithm [Baluja et al., 2008] (Chapter 4) the direction of these edges were inversed.

The graph was induced for a large amount of users where every user was influenced by many others. One of the advantages of our method is that keeping track of interactions counts between users avoids having to consider the distance between pairs of users. The graph induction algorithm runs in $O(n)$ time complexity (with n interactions). The disadvantage of this approach is that the strength of an incoming edge is determined relatively to every other incoming edge at a single node. This has a damping effect when the number of edges increases. A single strong connection can be suppressed as an increasing data volume introduces a large amount of weak connections.

Prior work exponentially weighs the edges to tighten strong connections (Equation 4). While this approach works for limited-size data sets, numerical instability became unavoidable due to the non-normalized nature of the edge weights (Section 3.3).

For this thesis we developed a new method to counter the damping effect of weak connections. Edge strengths are scaled using a quadratic function combined with a high-pass logistic filter.

$$y = \frac{x^{\beta}}{(1 + e^{(\alpha - x)})^{\beta}} \tag{20}$$

The choice of α determines the range influenced by the high-pass filter where a value of $\alpha = 6.0$ dampens values in lower ranges. Parameter β allows for the inflation of weights such that the distance between higher and lower values increases. For the purpose of this thesis the values 6.0 and 2.0 were assigned to these parameters respectively. Plots of the weighting function can be seen in Figure 3 and Figure 4 for different intervals.

2.1 Alternative interactions

Section 3.3 handles possible interactions on the social network that can be taken into account when modelling the graph. During experimentation all of these were evaluated. It was concluded that use of these features can be profitable in the long-term. These features raised additional questions and revealed issues that are outside of scope of this thesis. We briefly discuss these concerns here.

Sentiment analysis in tweets As shown by Pak and Paroubek [2010] the analysis of sentiment in tweets can provide good results. The creation of a specialized language model for our domain is out of scope of this study. A publicly-available language model distributed with Pattern[7], a web mining module developed by the Computational Linguistics & Psycholinguistics Research Center[8] at the University of Antwerp, was used. Performance of the model was measured by testing a hand-labeled sample of around 100 tweets and comparing these with the labels predicted by Pattern's sentiment analysis module. Unfortunately the prediction was only correct for around 50% of the validation set. This came as a surprise as they claim to achieve 75% accuracy on Dutch tweets [CLiPS]. One possible reason might be the colorful language used in tweets. Bermingham and Smeaton [2010] mention that the use of sentiment lexicons, such as SentiWordNet, are not useful in contexts similar to the one explored in this thesis.

Entity recognition in profile descriptions While many users use subjective and colorful language in their tweets, many of them presented non-ambiguous, objective descriptions of themselves in their profile descriptions. The use of these descriptions was investigated by applying some basic entity recognition by matching specially crafted regular expressions. This gave a 5% performance increase on average. The use of these features created a more complex model. Moreover the construction of these regular expressions were customized for the problem domain and could possibly lead to over-fitting. Therefore it was decided not to include these features in the final experiments.

[7] https://github.com/clips/pattern

[8] http://www.clips.ua.ac.be/pattern

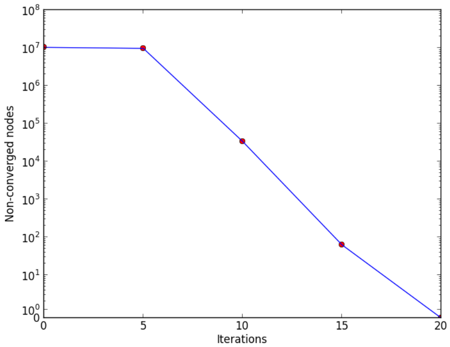


Fig. 5: Convergence of algorithm using the exploratory parameter configuration.

3 Exploratory parameters

In the first experiment parameters $w_{follows}$ and $w_{retweets}$ were set to one. The remaining parameters were set to zero as their semantics were unknown. Furthermore not much is known about any correlations between these variables. Therefore we only considered elementary features at first.

Figure 5 shows algorithm convergence. Most label distributions converge after fifteen iterations. After that the algorithm requires an additional five iterations for the remaining nodes. The full algorithm, including graph induction, usually takes a few of hours to complete. This depends on the size of the dataset, the available resources and the number of required iterations.

The confusion matrix for politically active users in $V_{selected}$[9] is shown in Table 3, along with the respective relevance measures in Table 4. See Section 4.2.3 for the definitions of macro- and micro-averaging of relevance measures. Remarkably we see that the results are dominated by a single class. Upon closer inspection it can be seen that this behavior is due to the betweenness centrality of hubs in the graph. These hubs are assigned a label in early iterations and then propagate the label to indecisive nodes in the network.

The contrast between label distribution $P(Y|V_{selected})$ (Figure 6) and distribution of classes c_i in $V_{selected}$ (Figure 7) confirms the dominance of the single class. Interestingly enough the label distribution expresses a rather divided opinion, while the distribution of classes is strongly skewed towards the dominant class.

A comparison between the relevance measures for $G_{selected}$ and $G_{observed}$ is given in Table 5. Performance when considering $G_{observed}$ is marginally better compared to that of $G_{selected}$.

[9] 340 out of a total 734 politically active users are contained in $V_{selected}$

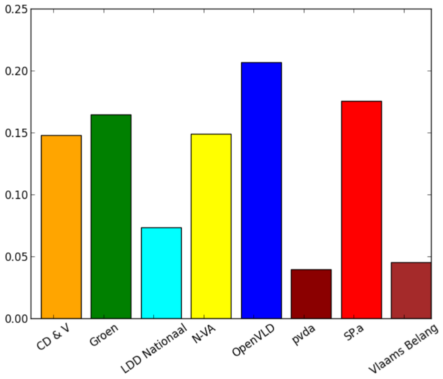


Fig. 6: Label distribution $P(Y|V_{selected})$ in the initial experiment.

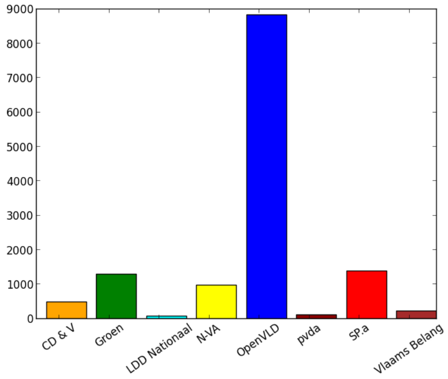


Fig. 7: Distribution of classes c_i in $V_{selected}$ (see Equation 1) in the initial experiment.

It is worth pointing out that remaining classes show good results in terms of precision (Table 4). In the next section different parameter configurations are evaluated in order to resolve the dominating class problem and consequently improve recall.

4 Optimized parameters

In order to increase model performance different parameter configurations (for parameters $w_{follows}$, $w_{followed}$, $w_{retweets}$, $w_{retweeted}$ and $w_{reciprocal}$) were considered. This was achieved by performing a parallelized grid search over the five-dimensional space $[0,1]^5$. Around a thousand different parameter configurations were used to classify a small training sample gathered from Twitter. The training sample consists out 226 users and 68 183 tweets. Contrary to the experiments described in this chapter, performance of a parameter configuration in the sample set was evaluated using its $V_{observed}$ set of users.

Cross-validation was performed by randomly splitting the validation set in two parts (respectively 60% and 40%). Recall, precision and F_1 were computed for both subsets. Table 6 shows the configurations that maximized the M-relevance (macro) for the training set. Each row in the table maximizes a different M-relevance measure. We considered each of these three configurations and selected the configuration that yielded best results on the full data set.

The experiment was repeated using the parameter configuration of Table 6 that optimizes F_1 (i.e. $w_{follows} = 0.4$, $w_{followed} = 0.3$, $w_{retweets} = 0.3$, $w_{retweeted} = 0.2$ and $w_{reciprocal} = 0.0$).

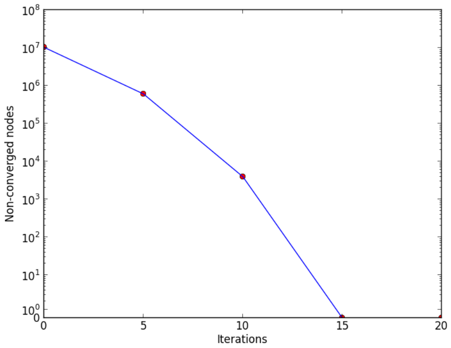


Fig. 8: Convergence of algorithm using the optimized parameter configuration.

Figure 8 shows algorithm convergence in the second experiment. Convergence is achieved for the majority of nodes after ten iterations. All nodes reach convergence after fifteen iterations. This is considerably faster than during the first experiment.

The confusion matrix (Table 7) shows improved performance over the previous experiment. From the table it seems that the problem of single-class domination is not present any longer. Relevance measures for the different classes confirm this trend in Table 8.

Unfortunately the class of party LDD exhibits disappointing performance compared to the other parties. This can explained by the inadequate amount of available validation data for this party. The confusion matrix (Table 7) shows that incorrectly labeled instances were assigned labels of parties closest in the political spectrum to LDD (see Figure 2), namely OpenVLD and N-VA. This observation raises additional questions regarding multi-label classification and the assessment of classification errors for similar classes.

Figure 9 and Figure 10 show the distributions $P(Y|V_{selected})$ and distribution of classes c_i in $V_{selected}$ respectively. In the first experiment we noticed that the class distribution was dominated by a single political party. In the second experiment the class distribution is no longer dominated by a single label.

The results in the second experiment look promising, but one should not forget that the validation set consists out of politically engaged individuals. In Section 4.2.4 a confidence measure based on Shannon entropy was defined. The measure assumes that neutral users carry a nearly uniform label distribution featuring high entropy, while involved participants have a stronger bias towards a particular class.

Figure 11 shows the cumulative distribution of label distribution polarity for both the total sample and the validation set. The distribution confirms prior

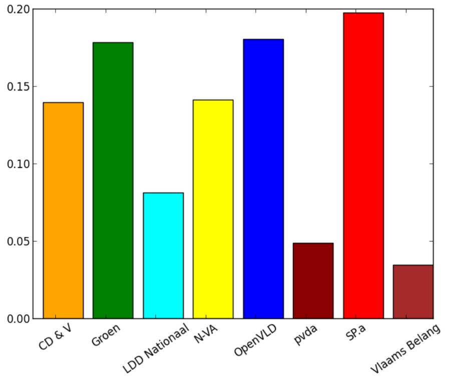


Fig. 9: Label distribution $P(Y|V_{selected})$ in the second experiment.

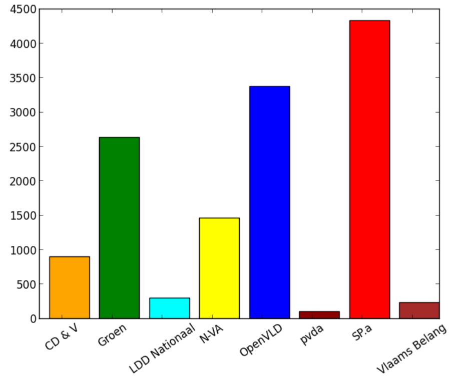


Fig. 10: Distribution of classes c_i in $V_{selected}$ (see Equation 1) in the second experiment.

suspicion of validation set bias towards the classification problem. The evolution of the macro-relevance measures over a varying discriminative threshold can be seen in Figure 12. The plot confirms the usefulness of label distribution entropy as a way to express confidence. Both precision and recall are maximized as the discriminative threshold increases.

In Figure 13 we show the distribution of classes for different values of polarity. The large population size with low label distribution polarity (Figure 13) raises additional questions regarding the value of the class distribution shown in Figure 10. We notice that most of the population taking the most prominent classes in Figure 13 generally have low label distribution polarity. While we observe good predictive performance on the validation set, the bulk of the population seems to expose neutral opinions. Given the good predictive performance and semi-supervised nature of the algorithm we assume this is due to a deficit of gathered data regarding these users.

To conclude this chapter a comparison between the relevance measures between $G_{observed}$ and $G_{selected}$ graphs is given in Table 9. Compared to the first experiment we can see that performance has improved for both graphs by approximately 10%. The difference with Table 5 of the first experiment is that in the second experiment $G_{selected}$ outperforms $G_{observed}$ in both recall and precision measures. A possible explanation of this is the use of inversed weights in the second experiment, resulting in a stronger connected graph when there is more data available. A visual representation of labeled graph $G_{selected}$ is given in Figure 14.

		Prediction outcome								
		CD & V	Groen	LDD Nationaal	N-VA	OpenVLD	pvda	SP.a	Vlaams Belang	
Actual value	CD & V	17	0	0	0	7	0	0	0	24
	Groen	0	56	0	0	5	0	0	0	61
	LDD Nationaal	0	0	1	2	2	0	0	0	5
	N-VA	0	0	0	24	11	0	0	0	35
	OpenVLD	1	0	0	0	60	0	1	0	62
	pvda	0	0	0	0	5	8	0	0	13
	SP.a	0	8	0	0	11	0	106	0	125
	Vlaams Belang	0	0	0	0	0	0	0	15	15
		18	64	1	26	101	8	107	15	

Table 3: Confusion matrix of classifications in $V_{selected}$ for the first experiment.

	Recall	**Precision**	$\mathbf{F_1}$
CD & V	70.83%	94.44%	0.8095
Groen	91.80%	87.50%	0.8960
LDD Nationaal	20.00%	100.00%	0.3333
N-VA	68.57%	92.31%	0.7869
OpenVLD	96.77%	59.41%	0.7362
pvda	61.54%	100.00%	0.7619
SP.a	84.80%	99.07%	0.9138
Vlaams Belang	100.00%	100.00%	1.0000
M (macro)	74.29%	91.59%	0.8204
μ (micro)	84.41%	84.41%	0.8441

Table 4: Relevance measures for individual classes and aggregated variants in $V_{selected}$ for the first experiment.

		Recall	**Precision**	$\mathbf{F_1}$
$\mathbf{G_{selected}}$	M (macro)	74.29%	91.59%	0.8204
	μ (micro)	84.41%	84.41%	0.8441
$\mathbf{G_{observed}}$	M (macro)	75.35%	93.72%	0.8354
	μ (micro)	87.08%	87.08%	0.8708

Table 5: Comparison of relevance measures between $G_{selected}$ and $G_{observed}$ for the first experiment.

Parameters	**Training set**			**Validation set**		
	Recall	**Precision**	$\mathbf{F_1}$	**Recall**	**Precision**	$\mathbf{F_1}$
$w_{follows} = 0.1$, $w_{followed} = 0.1$, $w_{retweets} = 0.2$, $w_{retweeted} = 0.2$, $w_{reciprocal} = 0.0$	74.55%	83.34%	0.7870	75.04%	85.11%	0.7976
$w_{follows} = 0.8$, $w_{followed} = 0.1$, $w_{retweets} = 0.2$, $w_{retweeted} = 0.2$, $w_{reciprocal} = 0.0$	71.93%	87.73%	0.7905	72.10%	83.26%	0.7728
$w_{follows} = 0.4$, $w_{followed} = 0.3$, $w_{retweets} = 0.3$, $w_{retweeted} = 0.2$, $w_{reciprocal} = 0.0$	73.69%	87.62%	0.8006	71.95%	82.75%	0.7698

Table 6: Different parameter configuration vectors that maximize macro-relevance measures in the training set.

		Prediction outcome								
		CD & V	Groen	LDD Nationaal	N-VA	OpenVLD	pvda	SP.a	Vlaams Belang	
Actual value	CD & V	23	1	0	0	0	0	0	0	24
	Groen	0	60	0	0	0	0	1	0	61
	LDD Nationaal	0	0	3	1	1	0	0	0	5
	N-VA	0	0	0	35	0	0	0	0	35
	OpenVLD	0	1	0	0	60	0	1	0	62
	pvda	0	4	0	0	0	8	1	0	13
	SP.a	0	2	0	0	0	0	123	0	125
	Vlaams Belang	0	0	3	1	0	0	0	11	15
		23	68	6	37	61	8	126	11	

Table 7: Confusion matrix of classifications in $V_{selected}$ for the second experiment.

	Recall	**Precision**	$\mathbf{F_1}$
CD & V	95.83%	100.00%	0.9787
Groen	98.36%	88.24%	0.9302
LDD Nationaal	60.00%	50.00%	0.5455
N-VA	100.00%	94.59%	0.9722
OpenVLD	96.77%	98.36%	0.9756
pvda	61.54%	100.00%	0.7619
SP.a	98.40%	97.62%	0.9801
Vlaams Belang	73.33%	100.00%	0.8462
M (macro)	85.53%	91.10%	0.8823
μ (micro)	95.00%	95.00%	0.9500

Table 8: Relevance measures for individual classes and aggregated variants in $V_{selected}$ for the second experiment.

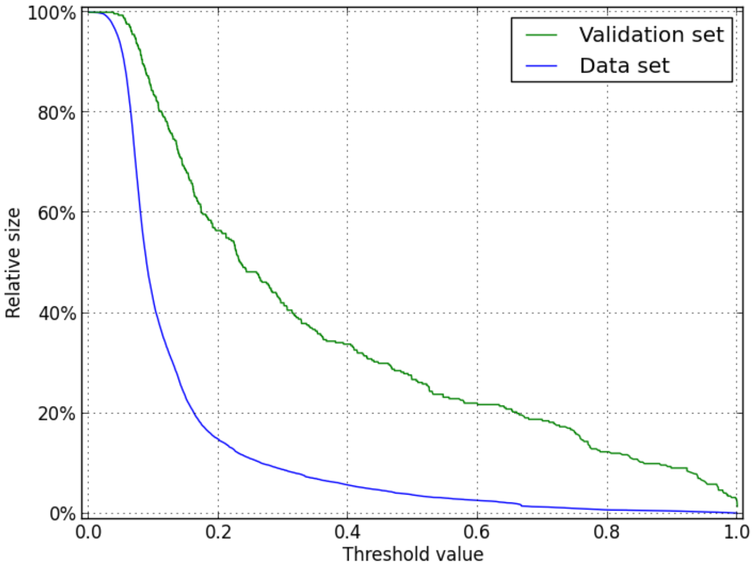


Fig. 11: Cumulative distribution of label distribution polarity.

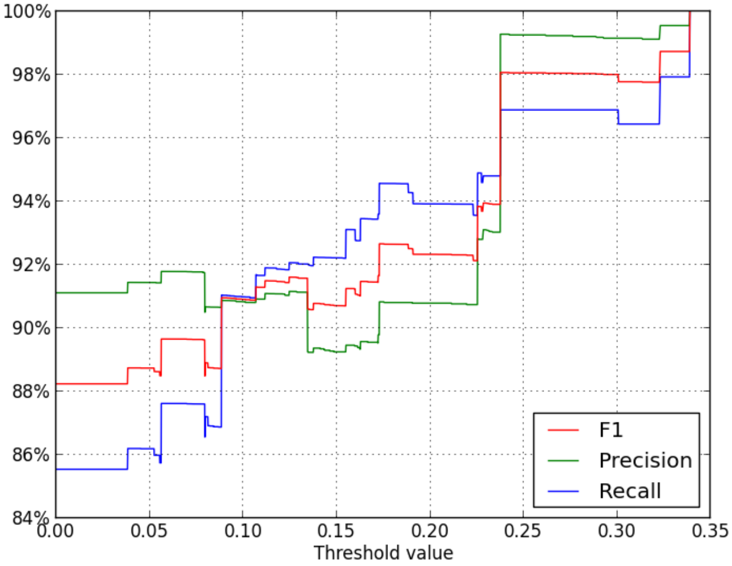


Fig. 12: Macro-relevance measures in function of varying polarity threshold.

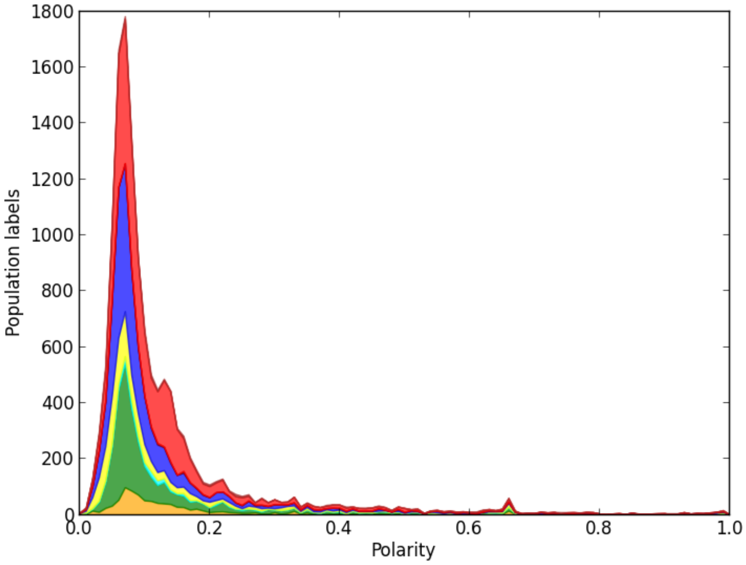


Fig. 13: Distribution of classes c_i in $V_{selected}$ for different polarity values. The graph hints that a large portion of users belonging to the dominant classes have a non-biased label distribution.

		Recall	**Precision**	$\mathbf{F_1}$
$\mathbf{G_{selected}}$	M (macro)	85.53%	91.10%	0.8823
	μ (micro)	95.00%	95.00%	0.9500
$\mathbf{G_{observed}}$	M (macro)	81.16%	90.15%	0.8542
	μ (micro)	94.13%	94.13%	0.9413

Table 9: Comparison of relevance measures between $G_{selected}$ and $G_{observed}$ for the second experiment.

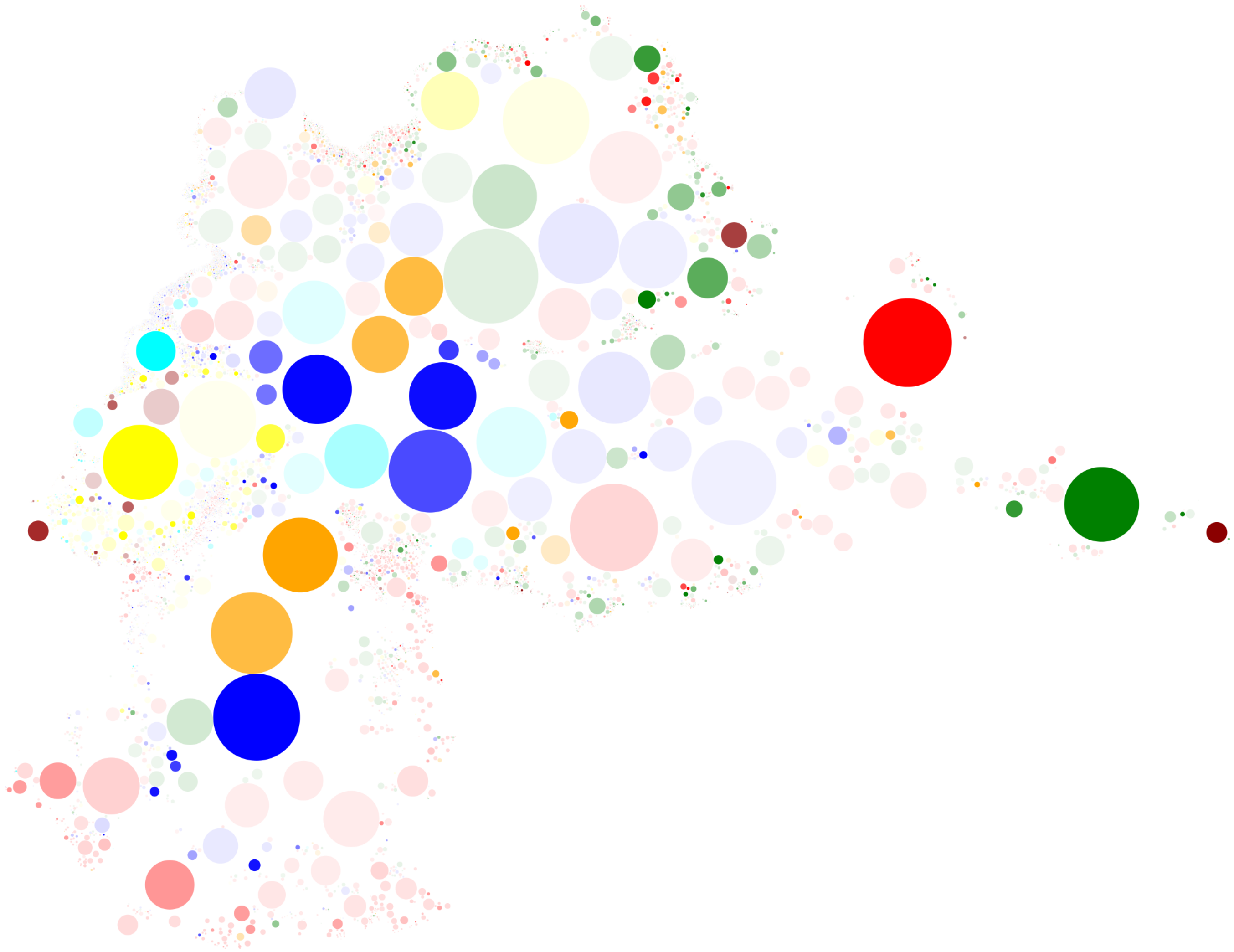


Fig. 14: Representation of the labeled graph $G_{selected}$. The coloring of a node represents the class that maximizes its label distribution (see Equation 1). Color opacity is set proportional to the confidence of the node's label distribution. The size of a node is proportional to its amount of outgoing edges.

Chapter 6

Future work

1 Multi-label classification

The method used in this thesis gives a distribution of labels $P(Y|i)$ for nodes $v_i \in V$. For the purpose of classification the class that maximizes the label distribution of a particular node (i.e. c_i) is selected as the final class (see Equation 1). While this approach works it might be sensible to consider different strategies.

One possibility is to consider the case where multiple labels can apply to a single instance, also known as *multi-label classification*. In terms of measuring performance it could be interesting to use a distance measure between the predicted and actual instance label, instead of a strict discriminative strategy when classes are similar or overlap.

2 Natural language processing

This thesis only considers tweet metadata. Their textual content was dismissed due to the absence of a good language model. The use of text contained in tweets as a way to quantify relations on social networks is a logical next step.

This involves applying natural language processing and statistical analysis methods on tweet content between two users, in order to determine the *likeness* or similarity between these users.

On one hand, the exchange of these messages can happen in an indirect or implicit manner over a longer period of time without structure in the ongoing discussion. On the other hand, there can be explicit threads of messages, usually occurring in a relatively short period of time. Both of these exchanges are interesting for determining similarities between users, which in time can be used to better model weights between nodes in social graphs.

3 Document classification and pattern mining

A possible application of semi-supervised learning is to obtain a larger set of labeled examples for training traditional classifiers, such as a text classifier.

In the previous chapter it was shown how graph classification can be applied on Twitter to infer the political atmosphere of social circles. Instances with high confidence could be used to train a model that classifies arbitrary social media by political affiliation. Classifier instances could be individual tweets labeled with the class of their author, or user descriptions.

Additional uses are in the field of pattern mining. Obtained labels could be used to find patterns in tweets belonging to a certain political affiliation. For example, it would be interesting to discover correlations between music preferences and political ideology.

Chapter 7

Conclusions

The recent uprise of social networks brings new challenges and opportunities in the field of data mining. Real-world use cases of these networks extend over multiple sociological, economical and political domains. The information contained within these networks is therefore interesting for various purposes. However, a lot of this information is unstructured and often provided in large quantities. The problem domain therefore lends itself to techniques for information extraction and knowledge discovery. We can use graph structures to model these networks and consequently apply techniques from the field of graph theory and similar fields to infer novel facts about social data.

In this thesis we focused on random walk-based methods for information extraction on social graphs. In particular we looked at determining the political climate of the social circles of Flemish users on Twitter. The official Twitter accounts of different Flemish political parties were taken as absorbing states in the underlying Markov chain. Consequently, every random walk on the graph was deemed to end at one of these parties. After evaluation we retrieved a probability distribution for every user in the data set. This distribution effectively indicates the probability that a random walk, starting at that particular user, terminates at each political party.

In order to accomplish this we gathered interaction data from Twitter and induced a graph based on this information. In order to determine the edge weights we focused on individuals following and retweeting one another. We evaluated different model configurations, edge weighting strategies and experimented with additional features. These features include user mentions in tweets, replies to tweets and even references to objects or mentions of hashtags in tweets. We concluded that, while these features are very valuable, their inclusion demands a reliable sentiment analysis language model. Due to the concise nature of tweets and often colorful language contained within them, models that work on traditional social media (web pages, blogs) are not guaranteed to give good results on microblogging content.

Additionally we considered applying a similar method on the descriptions contained within profiles. We noticed that these descriptions on average consist of neutral and objective language. Experiments extracting entity references from these descriptions showed good results. Eventually we decided against their use in order to avoid overfitting. Inclusion of these features raise more questions and could obfuscate results achieved by the random walk methods.

Different experiments showed that straight-forward unit weight configurations are a good start. We continued by performing a grid search on the space of parameters for a small subset of the full data set and discovered better performing parameter configurations. Additionally we noticed a problem when the size of the graph increased. Large quantities of weak edges would dampen the influence of strong edges. Different edge weighting strategies, such as exponential weighting and inversed weighting, were considered. While an exponential weighting strategy was used in prior work, we quickly ran into numerical issues. We opted for a mixture of a quadratic weighting function combined with a logistic function. Furthermore we noticed that the addition of self-loops to the graph model improved graph convergence. Additional experiments with newly gathered data sets confirmed the usefulness of these techniques.

Predictive performance of our methods was evaluated using a validation set of individuals associated with the different parties. This ground truth data was retrieved from lists of Twitter accounts published by the parties themselves. We observed satisfying results in the two experiments discussed in Chapter 5. The first and second experiment scored 0.82 and 0.88 in terms of macro F_1 score respectively.

In order to analyze the remaining data set population we looked at the polarity of their label distributions. For this we defined a measure based on Shannon entropy which quantifies the amount of information contained within a probability distribution. On one hand, a label distribution biased towards a single class has zero entropy and thus a high polarity value. On the other hand, a completely uniform label distribution has the highest possible entropy and therefore low polarity. We noticed that a large part of the population has a non-polarized label distribution and thus exhibits a neutral political atmosphere in their social circles. While this is certainly plausible, we believe that a larger amount of data could give more satisfying results.

To summarize, the following contributions were made:

- A strategy for retrieving an interesting sample from Twitter in the context of semi-supervised learning given a set of seed nodes (Section 3.2). In particular there was a focus on efficiently retrieving a data set that was most likely to receive a meaningful labeling.
- Analysis of the Twitter social network, how users on the network interact and the semantics of these interactions (Section 3.3).
- From gathered Twitter data a graph was induced based on social interactions on the social network (Section 3.3). Graph construction occurs in a scalable, tractable way such that it scales linearly with the data set size (Section 4.1). The social graph is represented by a sparse table of edge pairs and associated weights.
- An implementation of the Adsorption [Baluja et al., 2008] algorithm in the data-parallel Apache Crunch framework for execution on data processing

platforms such as Apache Hadoop, with various configuration tweaks, graph labeling analysis and post-processing tools for debugging and data visualization (Chapter 4).

- Improved algorithm performance and convergence speed when introducing dynamic self-loops in the graph (Section 4.1.2).
- A way to measure classification confidence by label distribution polarity based on Shannon entropy (Section 4.2.4).
- Application of the graph classification methods on Flemish Twitter users in order to determine the political climate of their social circles (Chapter 5). Experimental results with varying model parameters using a validation set of politically active users to measure classification performance showed high values for recall and precision measures. We explored the influence of different confidence thresholds on model performance.
- We showed that even with a small amount of information it is possible for machines to automatically derive privacy-sensitive facts from social networks (Chapter 5).

List of Figures

List of Tables

Bibliography

S. Asmussen. *Applied Probability and Queues*. Wiley, 1987.

Arik Azran. The rendezvous algorithm: Multiclass semi-supervised learning with markov random walks. In *In Proceedings of the 24th International Conference on Machine Learning*, 2007.

Shumeet Baluja, Rohan Seth, D. Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. Video suggestion and discovery for youtube: Taking random walks through the view graph. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 895–904, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-085-2. doi: 10.1145/1367497.1367618. URL `http://doi.acm.org/10.1145/1367497.1367618`.

Adam Bermingham and Alan F. Smeaton. Classifying sentiment in microblogs: Is brevity an advantage? In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1833–1836, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0099-5. doi: 10.1145/1871437.1871741. URL `http://doi.acm.org/10.1145/1871437.1871741`.

Smriti Bhagat, Graham Cormode, and S. Muthukrishnan. Node classification in social networks. *CoRR*, abs/1101.3291, 2011.

Danah Boyd, Scott Golder, and Gilad Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, HICSS '10, pages 1–10, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-3869-3. doi: 10.1109/HICSS.2010.412. URL `http://dx.doi.org/10.1109/HICSS.2010.412`.

David Recordon Brad Fitzpatrick. Thoughts on the social graph, 2007. URL `http://bradfitz.com/social-graph-problem/`. Visited on 18/05/2014.

Doug Braun. The evils of reciprocal following on twitter. URL `http://thedustpan.com/2010/03/evils-of-reciprocal-following/`. Visited on 05/01/14.

Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010. ISBN 0262514125, 9780262514125.

CLiPS. Belgian elections, june 13, 2010 - twitter opinion mining. `http://www.clips.ua.ac.be/pages/pattern-examples-elections`. Accessed: 2014-01-26.

Frances Perry Stephen Adams Robert R. Henry Robert Bradshaw Nathan Weizenbaum Craig Chambers, Ashish Raniwala. Flumejava: Easy, efcient data-parallel pipelines. `http://faculty.neu.edu.cn/cc/zhangyf/cloud-bigdata/papers/big%20data%20programming/FlumeJava-pldi-2010.pdf`, 2010.

Jef Dean and Sanjay Ghemawat. Mapreduce: Simplied data processing on large clusters. http://static.usenix.org/event/osdi04/tech/full_papers/dean/dean.pdf, 2004.

K. Deschouwer. *De stemmen van het volk: een analyse van het kiesgedrag in Vlaanderen en Wallonië op 7 juni 2009.* VUBPress, 2010. ISBN 9789054877356. URL http://books.google.co.uk/books?id=tQj2afpZOfwC.

Reinhard Diestel. *Graph Theory* {*Graduate Texts in Mathematics; 173*}. Springer-Verlag Berlin and Heidelberg GmbH & Company KG, 2000.

Jim Dougherty. Why reciprocal following on twitter does not matter. URL http://leaderswest.com/2012/01/05/you-dont-need-to-follow-me-but-i-appreciate-that-you-do/. Visited on 05/01/14.

Charles Duhigg. Campaigns mine personal lives to get out vote. URL http://www.nytimes.com/2012/10/14/us/politics/campaigns-mine-personal-lives-to-get-out-vote.html. Visited on 05/04/14.

William Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, January 1968. ISBN 0471257087. URL http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20{&}path=ASIN/0471257087.

The Apache Software Foundation. Apache crunch: Simple and efficient mapreduce pipelines. http://crunch.apache.org/, 2013.

M.S. Granovetter. The Strength of Weak Ties. *The American Journal of Sociology*, 78(6):1360–1380, 1973.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction.* Springer, 2 edition, 2008. URL http://scholar.google.com/scholar.bib?q=info:roqIsr0iT4UJ:scholar.google.com/&output=citation&hl=en&ct=citation&cd=0.

Gaby Hinsliff. Web 2.0: the new election superweapon. URL http://www.theguardian.com/politics/2010/apr/11/new-media-election-campaign. Visited on 04/01/14.

Philip N. Howard. The arab springs cascading effects. URL http://www.psmag.com/navigation/politics-and-law/the-cascading-effects-of-the-arab-spring-28575/. Visited on 04/01/14.

Shunsuke Ihara. *Information theory - for continuous systems.* World Scientific, 1993. ISBN 978-981-02-0985-8.

Frederic Jacobs. Do you want to know more about belgium? http://www.youtube.com/watch?v=Ceg6NQKHd70. Accessed: 2014-01-18.

R. Junco, G. Heiberger, and E. Loken. The effect of twitter on college student engagement and grades. *Journal of Computer Assisted Learning*, 27(2):119–132, 2011. ISSN 1365-2729. doi: 10.1111/j.1365-2729.2010.00387.x. URL http://dx.doi.org/10.1111/j.1365-2729.2010.00387.x.

Andreas M Kaplan and Michael Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business horizons*, 53(1):59–68, 2010.

Andreas M. Kaplan and Michael Haenlein. The early bird catches the news: Nine things you should know about micro-blogging. *Business Horizons*, 54 (2):105–113, March 2011. URL http://ideas.repec.org/a/eee/bushor/v54yi2p105-113.html.

Jemima Kiss. Facebook's 10th birthday: from college dorm to 1.23 billion users. 2014. URL http://www.theguardian.com/technology/2014/feb/04/facebook-10-years-mark-zuckerberg. Visited on 29/04/14.

Gueorgi Kossinets. Effects of missing data in social networks. *Social Networks*, 28:247–268, 2003.

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis: The good the bad and the omg! In Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts, editors, *ICWSM*. The AAAI Press, 2011. URL http://dblp.uni-trier.de/db/conf/icwsm/icwsm2011.html#KouloumpisWM11.

Raffi Krikorian. Introducing twitter data grants, 2014. URL https://blog.twitter.com/2014/introducing-twitter-data-grants.

P. F. Lazarsfeld and R. K. Merton. Friendship as a social process: a substantive and methodological analysis. In M. Berger, editor, *Freedom and Control in Modern Society*, pages 18–66. New York: Van Nostrand, 1954.

Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 631–636, New York, NY, USA, 2006. ACM. ISBN 1-59593-339-5. doi: 10.1145/1150402.1150479. URL http://doi.acm.org/10.1145/1150402.1150479.

Qing Lu and Lise Getoor. Link-based classification using labeled and unlabeled data. In *ICML Workshop on 'The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining'*, 2003.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001. doi: 10.1146/annurev.soc.27.1.415. URL http://arjournals.annualreviews.org/doi/abs/10.1146/annurev.soc.27.1.415.

Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *In Proceedings of the 5th ACM/USENIX Internet Measurement Conference (IMC07*, 2007.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012. ISBN 026201825X, 9780262018258.

Jennifer Neville and David Jensen. Iterative classification of relational data. In *Papers of the AAAI-2000 Workshop on Learning Statistical Models From Relational Data*. AAAI Press, 2000.

PuteriN.E. Nohuddin, Rob Christley, Frans Coenen, Yogesh Patel, Christian Setzkorn, and Shane Williams. Social network trend analysis using frequent pattern mining and self organizing maps. In Max Bramer, Miltos Petridis, and Adrian Hopgood, editors, *Research and Development in Intelligent Systems XXVII*, pages 311–324. Springer London, 2011. ISBN 978-0-85729-129-5. doi: 10.1007/978-0-85729-130-1_24. URL http://dx.doi.org/10.1007/978-0-85729-130-1_24.

Larry Page, Sergey Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web, 1998.

Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7.

Bo Pang and Lillian Lee. Opinion mining and sentiment analysis, January 2008. ISSN 1554-0669. URL http://dx.doi.org/10.1561/1500000011.

Alexandrin Popescul, Lyle H. Ungar, Steve Lawrence, and David M. Pennock. Towards structural logistic regression: Combining relational and statistical learning, 2002.

Paul Resnick and Hal R. Varian. Recommender systems. *Commun. ACM*, 40 (3):56–58, March 1997. ISSN 0001-0782. doi: 10.1145/245108.245121. URL http://doi.acm.org/10.1145/245108.245121.

Jeff J. Roberts. Typical twitter user is a young woman with an iphone & 208 followers. URL http://gigaom.com/2012/10/10/the-typical-twitter-user-is-a-young-woman-with-an-iphone-and-208-followers/. Visited on 05/01/14.

Jari Saramäki and JP Onnela. Structure and tie strengths in mobile communication networks. *Proc. Natl. Acad. Sci. (USA)*, page 7332, 2007.

Kurt Scholle. Follow you follow me increase twitter followers with reciprocal follows. URL http://website-roi-guy.com/76/twitter/. Visited on 05/01/14.

Kyle Siegrist. *Virtual Laboratories in Probability and Statistics*. University of Alabama in Huntsville, 2001. URL http://www.math.uah.edu/stat/. Visited on 04/01/14.

Martin Szummer and Tommi Jaakkola. Partially labeled classification with markov random walks. In *Advances in Neural Information Processing Systems*, pages 945–952. MIT Press, 2002.

The Apache Software Foundation. Apache hadoop. http://hadoop.apache.org/, 2005.

The Apache Software Foundation. Apache spark: Lightning-fast cluster computing. http://spark.apache.org/, 2013.

A. M. Turing. Computing machinery and intelligence. 59(236):433–460, October 1950. ISSN 0026-4423. URL http://turing.ecs.soton.ac.uk/browse.php/B/19;http://turing.ecs.soton.ac.uk/browse.php/B/9.

Twitter. Twitter api, a. URL https://dev.twitter.com. Visited on 11/01/14.

Twitter. Twitter api, b. URL https://dev.twitter.com/docs/entities. Visited on 11/01/14.

Twitter. Twitter api, c. URL `https://dev.twitter.com/docs/platform-objects/tweets`. Visited on 11/01/14.

S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications.* Cambridge University Press, 1994.

Eric W. Weisstein. Markov chain, a. URL `http://mathworld.wolfram.com/MarkovChain.html`. Visited on 02/01/14.

Eric W. Weisstein. Matrix power, b. URL `http://mathworld.wolfram.com/MatrixPower.html`. Visited on 03/01/14.

Eric W. Weisstein. Random walk, c. URL `http://mathworld.wolfram.com/RandomWalk.html`. Visited on 04/01/14.

Eric W. Weisstein. Stochastic matrix, d. URL `http://mathworld.wolfram.com/StochasticMatrix.html`. Visited on 02/01/14.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1220575.1220619. URL `http://dx.doi.org/10.3115/1220575.1220619`.

Yiming Yang, Sean Slattery, and Rayid Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18:219–241, 2002.

Shaozhi Ye, Juan Lang, and Felix Wu. Crawling online social graphs. In *Proceedings of the 2010 12th International Asia-Pacific Web Conference*, APWEB '10, pages 236–242, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4012-2. doi: 10.1109/APWeb.2010.10. URL `http://dx.doi.org/10.1109/APWeb.2010.10`.

Emma Young. Crisis puts a new face on social networking. URL `http://www.smh.com.au/federal-politics/crisis-puts-a-new-face-on-social-networking-20090210-83fk.html`. Visited on 04/01/14.

Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005. URL `http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf`.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions, 2003.

Appendices

Appendix A

Media coverage

On May 17, 2014 the University of Antwerp published a press release in collaboration with the Advanced Database Research and Modelling (ADReM) research group on the work performed in this thesis. The news was spread by the largest Flemish news agency (Belga) and picked up by the most notable Flemish news sites, such as *De Redactie*, *De Morgen*, *De Standaard* and others. Public broadcasting radio station *Radio 1* invited professor Bart Goethals on its morning show *Hautekiet*, named after the show's host, to explain the methods we applied.

Research group ADReM launched the website `twitterbrengtraad.be` (see Figure 15) simultaneously with the press release. On this website Twitter users were able to see the influence of the political parties in their own social circles. Users had the ability to make their own political atmosphere public by tweeting them. In addition they were able to provide feedback on the study, in particular we asked them if the political atmosphere of their circles corresponded with their personal political preference. Only users part of $V_{observed}$ (see Section 3.2) were able to query the political atmosphere of their Twitter social circles.

The results of the feedback study are out of scope of this work and may be published by the ADReM research group at a later time. To give a rough indication of the effectiveness of our method: more than half of the received feedback was positive.

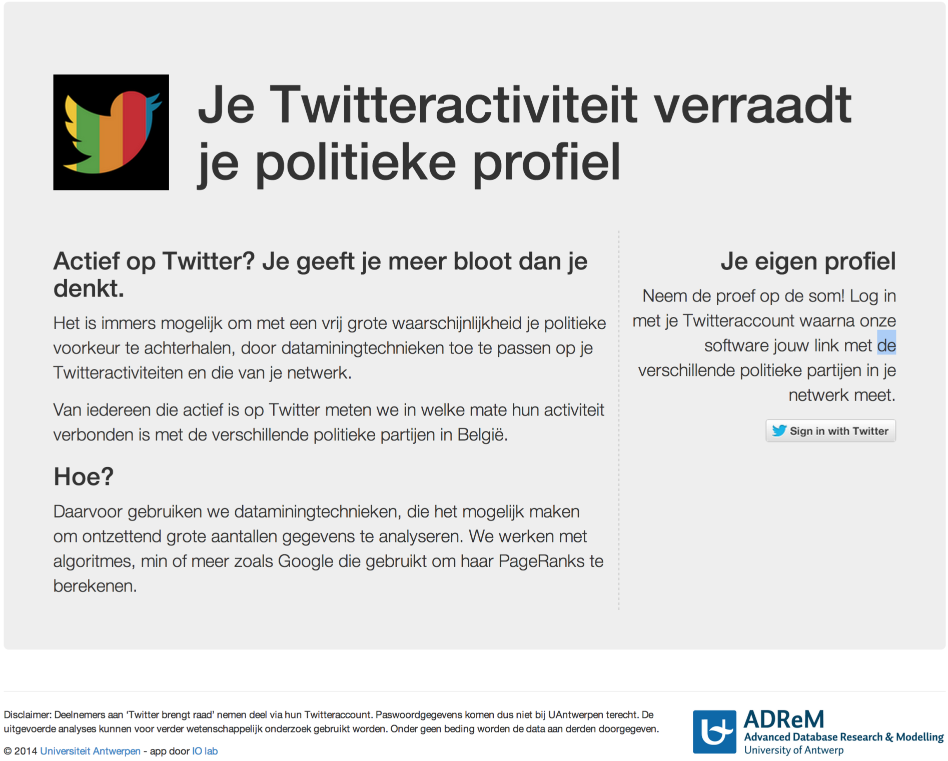
For the press release we refreshed the tweet dataset used in this work. We kept the same set of selected Twitter accounts as those selected in January 2014. We did not retrieve new follower relations. At the beginning of May we retrieved the 200 most-recent tweets from Twitter at that time and mixed those with the existing follower graph. All other parameters were kept the same as those in the second experiment (Section 5.4). Table 10 shows the confusion matrix of the classification. Results in terms of relevance performance can be seen in Table 11. As can be seen performance increased with this new mixed data set. As reported in the press release, we achieved a F_1 score of 0.94 on the validation set of politically active users.

		Prediction outcome								
		CD & V	Groen	LDD Nationaal	N-VA	OpenVLD	pvda	SP.a	Vlaams Belang	
Actual value	CD & V	24	0	0	0	0	0	0	0	24
	Groen	0	59	0	0	0	0	2	0	61
	LDD Nationaal	0	0	3	1	0	0	1	0	5
	N-VA	0	0	0	35	0	0	0	0	35
	OpenVLD	0	0	0	0	60	0	2	0	62
	pvda	0	0	0	0	0	12	1	0	13
	SP.a	0	2	1	1	0	0	121	0	125
	Vlaams Belang	0	0	0	0	0	0	0	15	15
		24	61	4	37	60	12	127	15	

Table 10: Confusion matrix of classifications in $V_{selected}$ for the dataset used by `twitterbrengtraad.be`.

	Recall	**Precision**	$\mathbf{F_1}$
CD & V	100.00%	100.00%	1.0000
Groen	96.72%	96.72%	0.9672
LDD Nationaal	60.00%	75.00%	0.6667
N-VA	100.00%	94.59%	0.9722
OpenVLD	96.77%	100.00%	0.9836
pvda	92.31%	100.00%	0.9600
SP.a	96.80%	95.28%	0.9603
Vlaams Belang	100.00%	100.00%	1.0000
M (macro)	92.83%	95.20%	0.9400
μ (micro)	96.76%	96.76%	0.9676

Table 11: Relevance measures for individual classes and aggregated variants in $V_{selected}$ for the dataset used by `twitterbrengtraad.be`.



Je Twitteractiviteit verraadt je politieke profiel

Actief op Twitter? Je geeft je meer bloot dan je denkt.

Het is immers mogelijk om met een vrij grote waarschijnlijkheid je politieke voorkeur te achterhalen, door dataminingtechnieken toe te passen op je Twitteractiviteiten en die van je netwerk.

Van iedereen die actief is op Twitter meten we in welke mate hun activiteit verbonden is met de verschillende politieke partijen in België.

Hoe?

Daarvoor gebruiken we dataminingtechnieken, die het mogelijk maken om ontzettend grote aantallen gegevens te analyseren. We werken met algoritmes, min of meer zoals Google die gebruikt om haar PageRanks te berekenen.

Je eigen profiel

Neem de proef op de som! Log in met je Twitteraccount waarna onze software jouw link met de verschillende politieke partijen in je netwerk meet.

Sign in with Twitter

Disclaimer: Deelnemers aan 'Twitter brengt raad' nemen deel via hun Twitteraccount. Paswoordgegevens komen dus niet bij UAntwerpen terecht. De uitgevoerde analyses kunnen voor verder wetenschappelijk onderzoek gebruikt worden. Onder geen beding worden de data aan derden doorgegeven.

© 2014 Universiteit Antwerpen - app door IO lab

Fig. 15: Website `twitterbrengtraad.be` launched simultaneously by the Advanced Database Research and Modelling research group with the press release.

1 Press release

The University of Antwerp released the following Dutch press release on its website and to various news agencies on May 17, 2014. The title of the press statement roughly translates to *"Activity on Twitter gives away your political preference"*.

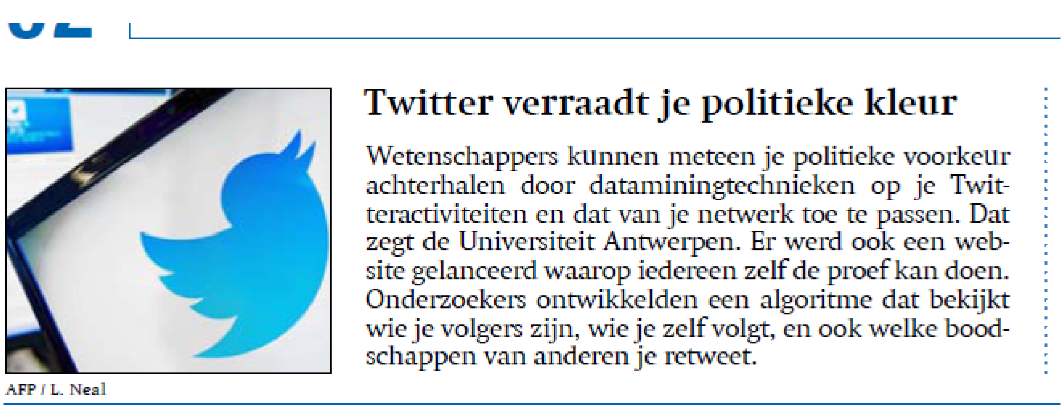
> **Twitteractiviteit verraadt je politieke profiel**
>
> Wie actief is op Twitter, geeft veel meer bloot over zichzelf dan hij of zij beseft. Onderzoekers van UAntwerpen kunnen je politieke voorkeur achterhalen door dataminingtechnieken toe te passen op je Twitteractiviteiten en die van je netwerk.
>
> In de aanloop naar de verkiezingen van 25 mei worden er heel wat peilingen en online stemtesten georganiseerd. De peilingen hebben tot doel de verkiezingsuitslag te voorspellen, de stemtesten geven je aan de hand van je antwoorden op een reeks stellingen een gefundeerd stemadvies. Informatici van UAntwerpen komen met een alternatieve stemtest op de proppen voor de actieve Twittergebruiker.
>
> Van iedereen die actief is op Twitter meten we in welke mate hun activiteit verbonden is met de verschillende politieke partijen in Vlaanderen, legt prof. Bart Goethals uit. Daarvoor gebruiken we dataminingtechnieken, die het mogelijk maken om grote hoeveelheden gegevens te analyseren. We werken met algoritmes, min of meer zoals Google die gebruikt om haar PageRanks te berekenen.
>
> Wie op Twitter zit, stuurt zelf korte uitspraken de wereld in, retweet de uitspraken van andere mensen of van instellingen en organisaties, volgt andere gebruikers op het online platform en wordt ook zelf gevolgd door anderen. Door al die gegevens kwantitatief te analyseren verzamel je per gebruiker een enorme hoeveelheid informatie, vertelt Christophe Van Gysel, die dit project in de context van zijn masterscriptie uitvoert.
>
> Wanneer we onze algoritmes loslaten op die gegevens, kunnen we het politieke profiel van een gebruiker vrij nauwkeurig meten. We deden een experiment met 734 accounts van Twittergebruikers die openlijk gelinkt zijn aan een specifieke partij. Onze analyse leverde enorm hoge juiste scores op. De accuraatheid is groter dan 95%.
>
> **Neem de proef op de som** Wat kan voor de partijgebonden Twitteraar, lukt ook voor de gebruiker die zich politiek niet meteen out op het sociale mediaplatform. Goethals: Een voorbeeld: als veel van de mensen in jouw netwerk een duidelijke voorkeur voor partij A hebben, omdat ze de officiële

Twitteraccount of politici uit die partij volgen, is dat al een eerste signaal dat ook jij partij A goedgezind kan zijn.

Twittergebruikers die de proef op de som willen nemen, surfen naar `www.twitterbrengtraad.be`. Op die projectsite log je in met je Twitteraccount, waarna de software jouw link met de verschillende partijen meet. Wie n week voor de verkiezingen nog niet weet hoe te stemmen, kan zich dus laten adviseren door zijn of haar online netwerk.

De computerwetenschappers van UAntwerpen willen met hun verkiezingsexperiment aantonen dat Twittergebruikers zonder het te beseffen veel over zichzelf blootgeven. Goethals: Wat je zelf de wereld instuurt in 140 tekens weet je natuurlijk wel, maar ook je andere activiteiten - mensen volgen, gevolgd worden, retweeten - zeggen heel wat over jouw profiel, zeker wanneer ook het profiel van alle mensen in je online netwerk mee in rekening wordt genomen.

2 Newspaper articles

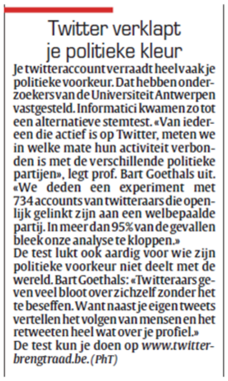


Twitter verraadt je politieke kleur

Wetenschappers kunnen meteen je politieke voorkeur achterhalen door dataminingtechnieken op je Twitteractiviteiten en dat van je netwerk toe te passen. Dat zegt de Universiteit Antwerpen. Er werd ook een website gelanceerd waarop iedereen zelf de proef kan doen. Onderzoekers ontwikkelden een algoritme dat bekijkt wie je volgers zijn, wie je zelf volgt, en ook welke boodschappen van anderen je retweet.

AFP / L. Neal

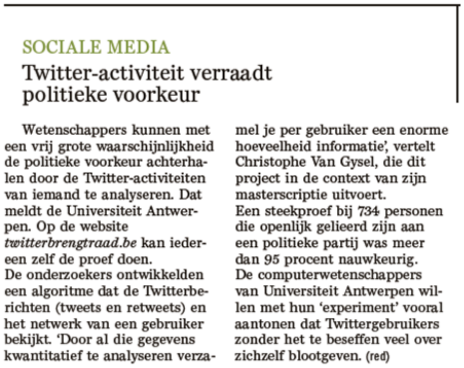
Fig. 16: Article in newspaper *Metro* (Monday May 19, 2014).



Twitter verklapt je politieke kleur

Je twitteraccount verraadt heel vaak je politieke voorkeur. Dat hebben onderzoekers van de Universiteit Antwerpen vastgesteld. Informatici kwamen zo tot een alternatieve stemtest. «Van iedereen die actief is op Twitter, meten we in welke mate hun activiteit verbonden is met de verschillende politieke partijen», legt prof. Bart Goethals uit. «We deden een experiment met 734 accounts van twitteraars die openlijk gelinkt zijn aan een welbepaalde partij. In meer dan 95% van de gevallen bleek onze analyse te kloppen.»
De test lukt ook aardig voor wie zijn politieke voorkeur niet deelt met de wereld. Bart Goethals: «Twitteraars geven veel bloot over zichzelf zonder het te beseffen. Want naast je eigen tweets vertellen het volgen van mensen en het retweeten heel wat over je profiel.»
De test kun je doen op *www.twitterbrengtraad.be*. *(PhT)*

Fig. 17: Article in newspaper *Het Laatste Nieuws* (Monday May 19, 2014).



SOCIALE MEDIA

Twitter-activiteit verraadt politieke voorkeur

Wetenschappers kunnen met een vrij grote waarschijnlijkheid de politieke voorkeur achterhalen door de Twitter-activiteiten van iemand te analyseren. Dat meldt de Universiteit Antwerpen. Op de website *twitterbrengtraad.be* kan iedereen zelf de proef doen.
De onderzoekers ontwikkelden een algoritme dat de Twitterberichten (tweets en retweets) en het netwerk van een gebruiker bekijkt. 'Door al die gegevens kwantitatief te analyseren verzamel je per gebruiker een enorme hoeveelheid informatie', vertelt Christophe Van Gysel, die dit project in de context van zijn masterscriptie uitvoert.
Een steekproef bij 734 personen die openlijk gelieerd zijn aan een politieke partij was meer dan 95 procent nauwkeurig.
De computerwetenschappers van Universiteit Antwerpen willen met hun 'experiment' vooral aantonen dat Twittergebruikers zonder het te beseffen veel over zichzelf blootgeven. (red)

Fig. 18: Article in newspaper *De Standaard* (Monday May 19, 2014).